

Министерство образования и науки РФ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Уральский государственный педагогический университет»  
Институт математики, физики, информатики и технологий  
Кафедра информационно-коммуникационных технологий в образовании

# ТЕХНОЛОГИЯ ПОДГОТОВКИ АДАПТИВНОГО ДИЗАЙНА ДЛЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

*Выпускная квалификационная работа  
бакалавра по направлению подготовки  
09.03.02 – Информационные системы и технологии*

Исполнитель: студентка группы ИСИТ-1501  
Института математики, физики, информатики  
и технологий  
Колташёва Д.Д.

Руководитель: к.п.н., доцент кафедры ИКТО  
Сардак Л.В.

Работа допущена к защите  
« 20 » \_\_\_\_\_ мая 2019 г.  
Зав. кафедрой \_\_\_\_\_

Екатеринбург – 2019

## Оглавление

<b>ВВЕДЕНИЕ.....</b>	<b>3</b>
<b>ГЛАВА 1. АНАЛИЗ ПОНЯТИЯ АДАПТИВНЫЙ ДИЗАЙН И ТЕХНОЛОГИЙ ЕГО ПРОЕКТИРОВАНИЯ.....</b>	<b>5</b>
1.1 ДИЗАЙН ИНТЕРФЕЙСА ИНФОРМАЦИОННОЙ СИСТЕМЫ.....	5
1.2 ТЕХНОЛОГИИ ПРЕДСТАВЛЕНИЯ РУКОВОДСТВА РАЗРАБОТЧИКА.....	11
1.3 ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ ИНФОРМАЦИОННОЙ СИСТЕМЫ (ТЕХНОЛОГИИ, ПРОГРАММНОГО ПРОДУКТА И ПР.) «ТЕХНОЛОГИЯ ПРОЕКТИРОВАНИЯ АДАПТИВНОГО ДИЗАЙНА» .....	17
<b>ГЛАВА 2. РЕАЛИЗАЦИЯ АДАПТИВНОГО ДИЗАЙНА ПО ТЕХНОЛОГИИ MATERIAL DESIGN ПРЕДСТАВЛЕННОГО В ФОРМАТЕ РУКОВОДСТВА ПОЛЬЗОВАТЕЛЯ.....</b>	<b>22</b>
2.1 СТРУКТУРА РУКОВОДСТВА ПОЛЬЗОВАТЕЛЯ ПО РЕАЛИЗАЦИИ АДАПТИВНОГО ДИЗАЙНА ПО ТЕХНОЛОГИИ MATERIAL DESIGN.....	22
2.2 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ «ТЕХНОЛОГИЯ ПРОЕКТИРОВАНИЯ АДАПТИВНОГО ДИЗАЙНА» .....	26
2.3 РЕЗУЛЬТАТЫ АПРОБАЦИИ .....	51
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>52</b>
<b>СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ.....</b>	<b>53</b>

## **Введение**

В современном мире сложно представить человека, у которого нет мобильного телефона. Мобильные устройства стали универсальными помощниками для людей в самых разных областях знаний, начиная с повседневных задач, как например пробуждение по будильнику в установленное время, и заканчивая сложными математическими расчетами в специальных, узконаправленных программах. По данным независимых исследователей, современный человек проводит около 5 часов в день, используя мобильный телефон [23]. Вследствие чего, значимыми становятся дизайн-эргономические показатели отображаемой на экране информации.

Одной из задач разработчиков мобильных приложений является проектирование и реализация привлекательного интерфейса с оптимальным исходным кодом. В команду разработчиков входят дизайнеры, которые решают задачу подготовки прототипов конфигурации и макетов будущего приложения на основе дизайн эргономических показателей: простые и понятные сценарии работы с приложением, привлекательный внешний вид, масштабируемость дизайна для различных диагоналей смартфонов. Все эти показатели определяют адаптивность дизайна.

Под адаптивностью понимают способность дизайна обеспечивать корректное отображение интерфейса информационной системы на разных устройствах с разными операционными системами. Целью адаптивного (гибкого) дизайна является оптимальная разработка и единое отображение дизайна интерфейса на любых устройствах.

Продукт разработки – технология проектирования адаптивного дизайна.

Цель работы – разработать и описать технологию проектирования адаптивного дизайна.

Задачи:

1. Произвести анализ информационных источников по теме проектирования интерфейсов для мобильных устройств.

2. Рассмотреть способы представления и описания информационных технологий в справочной форме.
3. В соответствии с техническим заданием провести разработку «Технологии проектирования адаптивного дизайна»
4. Провести апробацию.

# **Глава 1. Анализ понятия адаптивный дизайн и технологий его проектирования**

## **1.1 Дизайн интерфейса информационной системы**

Понятие «интерфейс» имеет множество определений в разных сферах, но в области информационных технологий этот термин обозначает комплекс программных и технических средств, посредством которых осуществляется взаимодействие пользователя с информационной системой [9].

Программный интерфейс – это система унифицированных связей, предназначенных для обмена информацией между компонентами вычислительной системы. Программный интерфейс реализует процедуры, регламентирующие поведение сущностей по умолчанию. Таким образом реализации интерфейса будут унифицированными, но в то же время различными между собой.

Аппаратный интерфейс – комплекс устройств, таких как система шин, разъемы, которые преобразовывают сигналы и передают их от одних компонентов технических средств другим [24, 28]. От физических характеристик аппаратного интерфейса зависит быстродействие и надежность всей программно-аппаратной системы.

Интерфейс передачи данных обеспечивает взаимодействие всех элементов информационной системы, которые участвуют в процессе передачи данных. Примерами интерфейсов передачи данных являются следующие интерфейсы: Ethernet, USB, IrDA, HDMI, Bluetooth и Wi-Fi.

Интерфейс пользователя (user interface) – это набор программных и аппаратных средств, обеспечивающий взаимодействие пользователя с программно-аппаратными компонентами информационной системы с помощью диалогов [10]. Диалоги – это нормированный процесс обмена информацией между пользователем и устройством, который направлен на достижение совместных результатов. Процесс обмена происходит за счет ввода/вывода сообщений на экран устройства.

История развития пользовательских интерфейсов. В 50-е годы прошлого столетия появились первые решения взаимодействия пользователя с устройством в виде дополнительных печатных устройств, напоминающих современную клавиатуру. Первые печатные устройства реализовывали построчный ввод информации [23]. Взаимодействие происходило только между специально обученными операторами и устройствами, с помощью языка управления заданиями, в котором управляющие команды определяли режимы работы с программой и выполняли разработанные процедуры. Режимы работы с программой представляли собой только режим программиста, где выводился весь исходный код и режим оператора, который должен обладать специальными навыками работы с программой, так как пользовательского интерфейса вовсе не существовало.

Следующий этап эволюции интерфейсов приходится на период с начала 60-х годов до начала 70-х годов. В данном этапе различия с первым этапом не значительные, так как взаимодействие операторов с компьютером также производилось с помощью устройств ввода, но добавились специальные команды с параметрами [30]. Далее развитие продолжилось в начале 70-х годов в научно-исследовательской лаборатории компании Xerox. Компания разработала первый графический пользовательский интерфейс для работы на растровых графических сетевых рабочих станциях. Первые интерфейсы реализовывали тип WIMP-интерфейсов или GUI – graphical user interface – графический интерфейс пользователя. В аббревиатуре WIMP – windows, icons, menu, pointing device содержатся названия основных интерактивных сущностей, таких как окно, пиктограмма и меню, используемых при построении интерфейса, а также WIMP-интерфейсы использовали первые примитивные позиционирующие устройства типа мышь. В дальнейшем именно этот тип интерфейса был усовершенствован компанией Apple в первой линейке персональных компьютеров Macintosh, и компанией Windows [31].

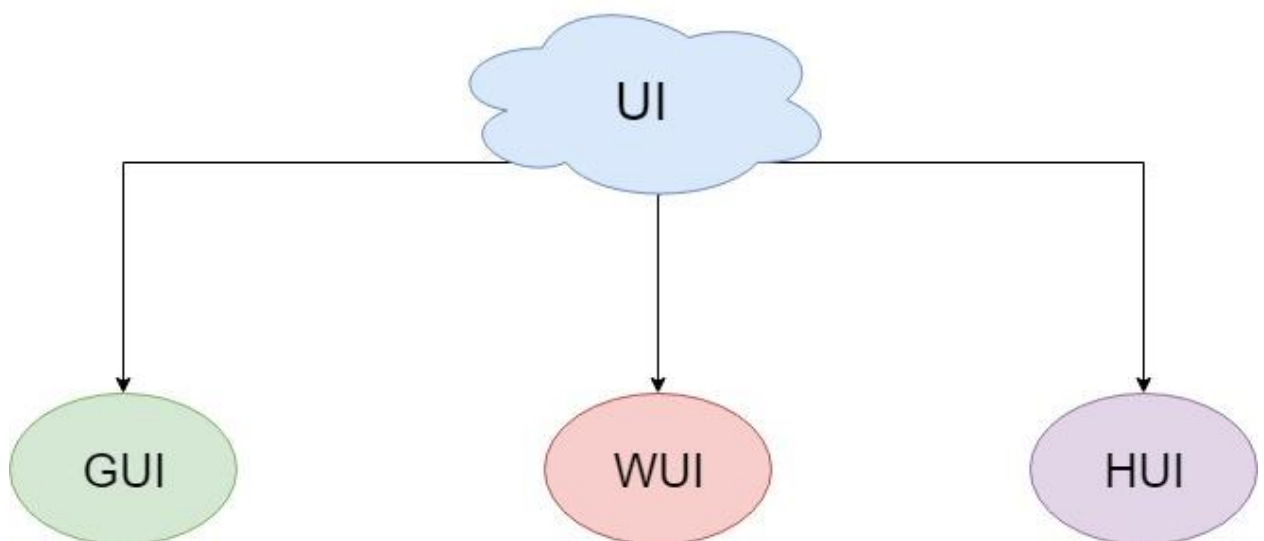
На сегодняшний день современные персональные компьютеры компании Windows реализуют тип WIMP-интерфейсов, но уже с использованием современных интерфейсных инструментов для создания более реалистичного отображения данных на экране устройства [32]. Современные интерфейсы активно используют большую палитру цветов, а также отличительной чертой стала доступность для широкого круга разработчиков наличие большого количества программного обеспечения для разработки WIMP-интерфейсов.

Основной задачей, решаемой пользователем при работе с пользовательским интерфейсом, является манипуляции с данными в программном обеспечении, поэтому интерфейс программного обеспечения пользователя должен соответствовать основным свойствам, которые гарантируют комфортное взаимодействие пользователя с информационной системой [26].

Свойства пользовательского интерфейса:

- Функциональность интерфейса обозначает высокую эффективность каждой операции пользователя. Операции должны быть максимально информативными, практичными и доступными для пользователей с разными уровнями опыта использования информационных систем.
- Естественность интерфейса подразумевает реализацию привычных для пользователя решений задач, то есть результаты работы после выполненной задачи, должны соответствовать реальным представлениям пользователя о задачах и не требовать дополнительных пояснений. Для этого достаточно использовать привычную для данной области систему терминов и обозначений.
- Дружественность предполагает, что интерфейс должен подстраиваться под пользователя, то есть пользователь должен ощущать, что он управляет процессом. В случае возникновения ошибок интерфейс должен оповещать пользователя и исключать возможность сценариев работы с критическими ошибками.

- Адаптивность интерфейса описывает его свойство гибкости экрана и функций. Так функции приложения должны подстраиваться под пользователей разного уровня подготовки использования информационных систем. Подразумевает возможность изменения структуры диалога и входных данных. Гибкость экрана подразумевает изменение внешнего вида приложения в зависимости от типа устройства, типа операционной системы, ориентации экрана и прочее.
- Обратная связь – в интерфейсе должен быть реализован отклик на каждое действие пользователя. Приложение должно реагировать на все изменения и использовать осмысленную анимацию для представления происходящих процессов. Так, например при загрузке каких-либо данных в интерфейсе привычно использование прокручивающихся анимаций, которые временно блокируют функции приложения на время загрузки файлов.



*Рис. 1. Классификация пользовательских интерфейсов.*

Существует три основных вида пользовательских интерфейсов, которые используются при проектировании информационных систем: графический интерфейс пользователя, веб-интерфейс пользователя, интерфейс пользователя мобильных устройств (Рис. 1).



Графический интерфейс пользователя (GUI – graphical user interface) – определяется как стиль взаимодействия пользователя с компьютером с помощью фундаментальных компонентов интерфейса – окно, пиктограмма, меню и указатель. Основные свойства GUI-интерфейса: поддержка работы с указателем типа мышь с возможностью непосредственного манипулирования, использование графики.

Веб-интерфейс пользователя (WUI – web user interface) – стиль взаимодействия пользователя с компьютером, отображаемый в иерархическом стиле. Общий вид веб-интерфейса представляет собой окно, навигация в котором осуществляется с помощью приложений в виде текстовых или визуальных гиперссылок [11]. В зависимости от структуры гиперссылок приложения, навигация во всем веб-интерфейсе приводит к представлению веб-страниц в одном окне каждой в линейном или нелинейном стиле. Основные особенности: вся информация приложения отображается в одном окне – браузере, в клиентской области отсутствует отображение пиктограмм.

Интерфейс пользователя мобильных устройств (HUI – human user interface) – специальный стиль взаимодействия пользователя с карманным мобильным устройством. Является подмножеством графического пользовательского интерфейса, от которого унаследовал основные элементы. Отличается размерами экранов устройств, способами представления на экранах информации и вводом данных, так как на мобильных устройствах зачастую используется сенсорный экран, и пользователи используют жестикуляционный ввод данных.

Сравним представленные интерфейсы по следующим характеристикам: основные элементы, которые используются для создания интерфейса, особенности интерфейсов, навигация по интерфейсу, дополнительные компоненты, используемые для создания интерфейса и особенности ввода данных.

*Таблица 1.*  
*Сравнение пользовательских интерфейсов.*

<b>Характеристики</b>	<b>Графический интерфейс пользователя</b>	<b>Веб-интерфейс пользователя</b>	<b>Интерфейс пользователя мобильных устройств</b>
Обозначение	GUI – graphical user interface	WUI – web user interface	HUI – human user interface
Основные элементы	Окно, пиктограмма, указатель, меню.	Окно, указатель	Окно, меню, пиктограмма
Особенность	Возможность оформления экрана приложения	Использование гиперссылок, отсутствие меню, пиктограмм	Отображение на экране только одного объекта.
Навигация	Полноценный функционал	Базовые функции: вперед, назад, гиперссылки	Базовые функции: вперед, назад.
Компоненты	Навигационные панели, диалоговое окно, полоса прокрутки, флажок.	Баннеры, навигационные панели, различные гиперссылки, браузер	Отличный функционал жестикуляционного стиля
Ввод данных	Указатель, клавиатура	Указатель, клавиатура	Жестикуляционный стиль ввода на сенсорном экране

Определим основные термины пользовательских интерфейсов:

- Окно – область устройства отображения, которая используется для представления объектов, взаимодействия с ними, представляет информацию об объектах и выполняет действия, нацеленные на работу с выбранным объектом.
- Пиктограмма (иконка) – область устройства отображения, предназначенная для наглядного представления объекта. Главным свойством пиктограммы является использование графических символов, необходимых для представления объектов, а также для возможности манипуляций с ними. Важнейшей операцией с пиктограммой является команда «Открыть», которая открывает окно с детализированной информацией о выбранном объекте.
- Меню представляет собой упорядоченный набор альтернатив, для предоставления выбора пользователя. Обычно в наборе альтерна-

тив предлагается выбор названий пользовательских команд, обозначающих действия над выбранным объектом. Существует несколько типов меню: каскадное меню, выпадающий список, всплывающее меню и строковое меню. Все эти типы отличаются способом отображения информации на экране.

- Указатель – графический вид отображения координатно-указательного устройства, предназначено для взаимодействия пользователя с информационной системой, управляя которым пользователь может совершать манипуляции для полноценной работы в интерфейсе.
- Баннер – это визуальный заголовок, который отображается вверху веб-страницы.
- Навигационная панель представляет собой список вариантов, состоящий из гиперссылок. Осуществляет быстрые переходы по информации на веб-страницах.
- Гиперссылка – это вариант выбора, отображающий информацию в новом окне или в текущем окне.
- Браузер является специальной программой для корректного отображения с форматов веб-страниц и работы с ними. Основные элементы: заголовок и навигационная панель.

Таким образом при использовании интерфейса пользователя мобильных устройств на первый план встает свойство адаптивности интерфейса, так как у мобильных устройств размеры экрана меньше, чем размеры экранов ПК. Для полноценного использования информационной системы на мобильном устройстве необходимо изменение внешнего вида программного продукта.

## **1.2 Технологии представления руководства разработчика**

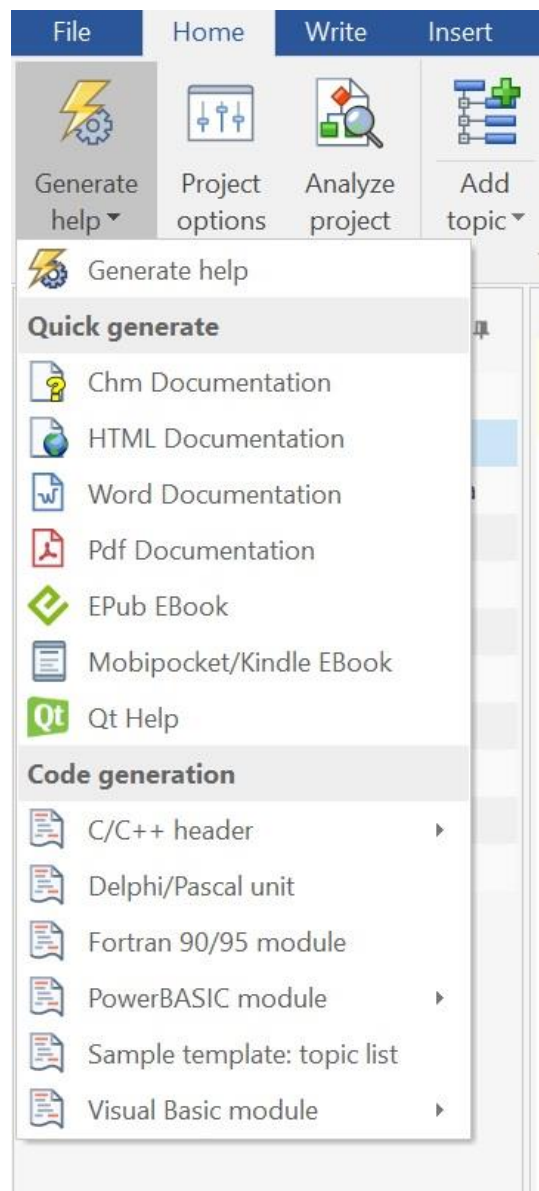
Для разработки информационной технологии «Технология проектирования адаптивного дизайна» были отобраны несколько программных продуктов,

которые удовлетворяют критериям, способствующим проектированию технологии. Критерии для программных продуктов:

- Возможность создания справочных материалов;
- Возможность контекстного поиска в разработанных материалах;
- Экспортирование материалов в разных форматах;

Программный продукт «Help’N’Doc» разработан для создания любых электронных документов в различных форматах. Преимущества программы:

- ✓ Возможность экспортирования электронных документов в форматах: chm, html, word, pdf, epub, mobipocket, qt help (Рис. 2).
  - Chm – формат веб-документов, представляющих справочную информацию. Часто в данном формате представлены электронные книги.
  - Html – формат онлайн документов.
  - Word – экспорт документов в формат для программы MS Word.
  - Pdf – формат электронных документов предназначенный для представления электронного документа в полиграфическом виде.
  - Epub – специальный файловый формат, разработанный для хранения электронных документов и книг. Создан Международной Ассоциацией Цифрового Книгоиздания (International Digital Publishing Association).
  - Mobipocket – специальное разрешение созданное для программы Mobipocket Reader. Формат хранит в себе электронные документы и книги.
  - Qt help – формат разработанный для создания специальных справочных документов.



*Рис. 2. Экспорт документов в программе «Help 'N' Doc»*

- ✓ Гибкая настройка представления документа и форматирования.
- ✓ Возможность создания элементов для дальнейшего использования. Например, создание переменной с названием, описываемой программы. После создания переменной, вам необходимо вставить её, а значение подставится автоматически.
- ✓ Выделение зарезервированных структур из разных языков программирования, таких как: AWK, C#, C++, CSS, COBOL, Eiffel, Fortran, Haskell, HTML, INI, ISS, Java, JavaScript, JSON, LLVM, Modelica, MS-DOS Batch,

Object Pascal, Perl, PHP, Python, Ruby, SQL, Tcl/Tk, TeX, UNIX Shell Script, Unreal, Visual Basic, Web IDL, x86 Assembly, XML.

- ✓ Интерактивная проверка орфографии.

Недостатками программы «Help’N’Doc» является отсутствие русской локализации в программе.

Программный продукт «Dr. Explain» предназначен для быстрого создания справочных документов, руководств пользователя, пособий и технических документаций к программному обеспечению или информационно-техническим системам [2, 3]. Преимущества программного продукта:

- ✓ Возможность экспортирования файлов в форматах html, chs, rtf, pdf. Приложение автоматически генерирует все выбранные форматы и предоставляет редактор для каждого из предложенных форматов;
  - Html – формат онлайн документов. В редакторе формата создается копия документа в виде онлайн документа. В документ автоматически вставляется навигационная панель с гиперссылками по оглавлению (Рис. 3).

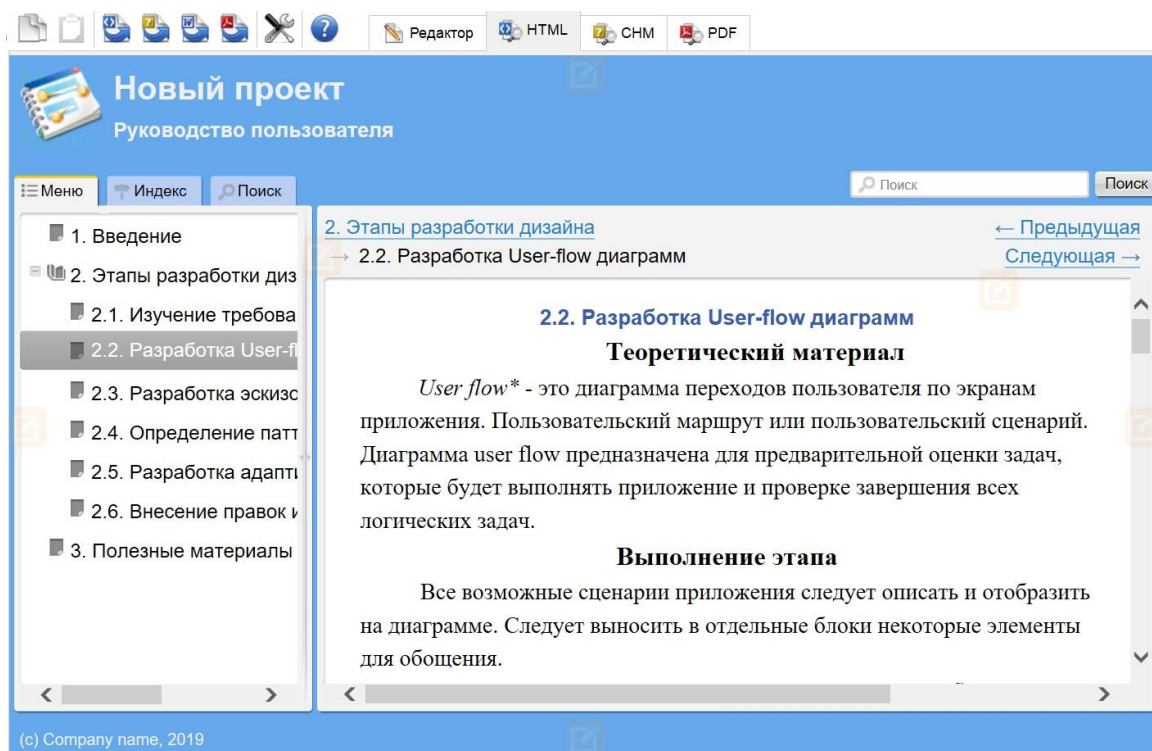


Рис. 3. Редактор html-документов.

- Chs – специальный формат справочных документов для операционной системы Windows (Рис. 4).

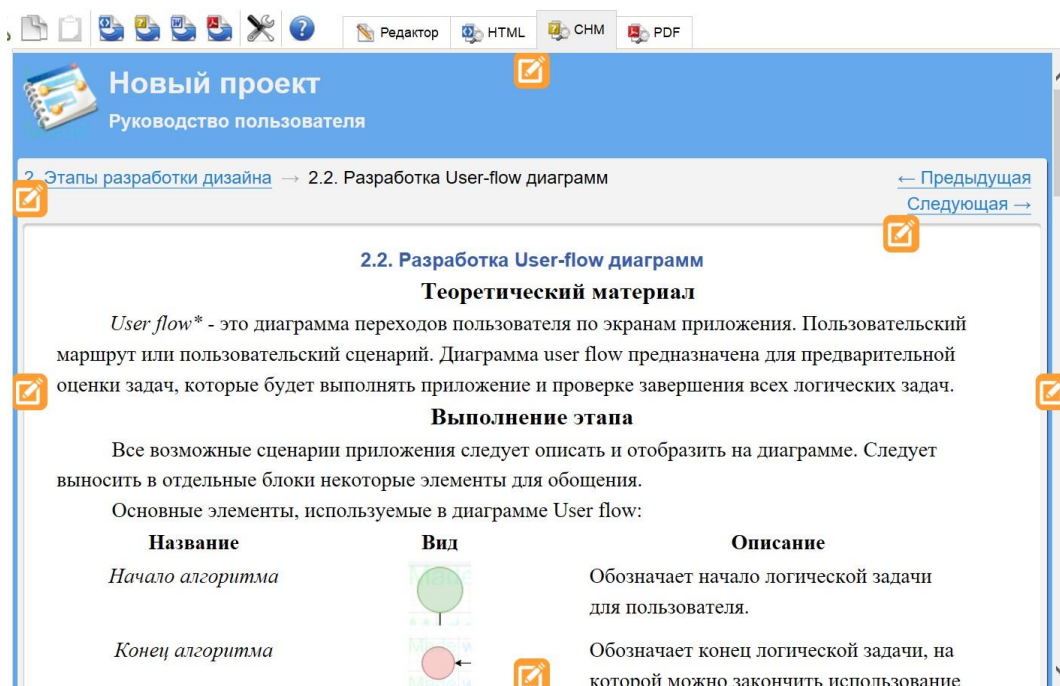


Рис. 4. Редактор chs-документов.

- Rtf – формат текстовых документов для специального форматирования документов (Рис. 5).

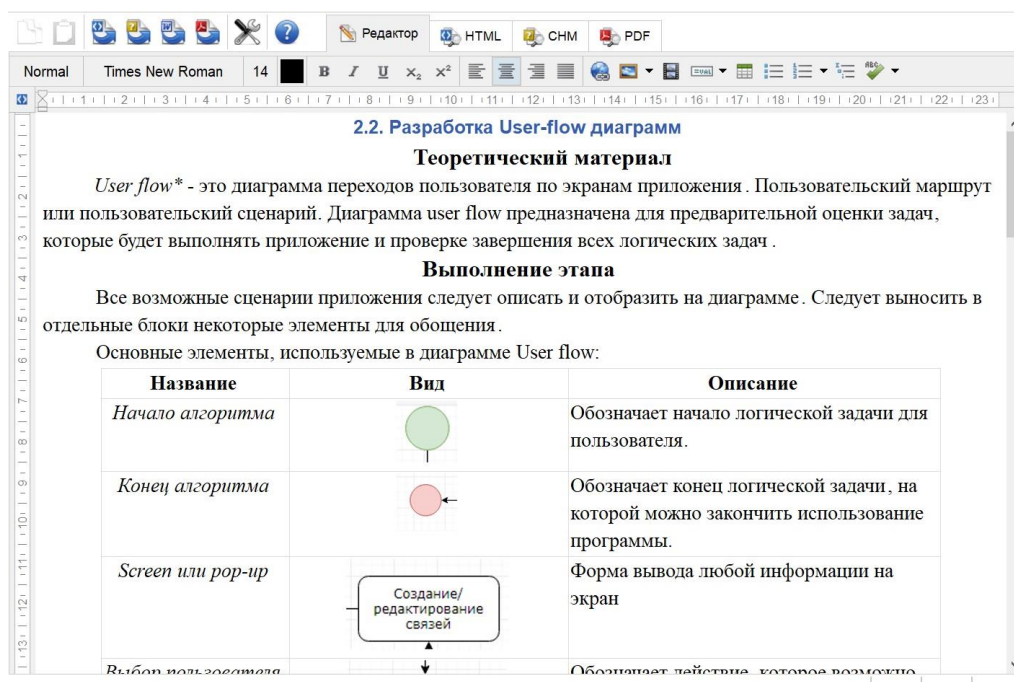


Рис. 5. Редактор rtf-документов.

- Pdf – формат электронных документов предназначенный для представления электронного документа в полиграфическом виде (Рис. 6).

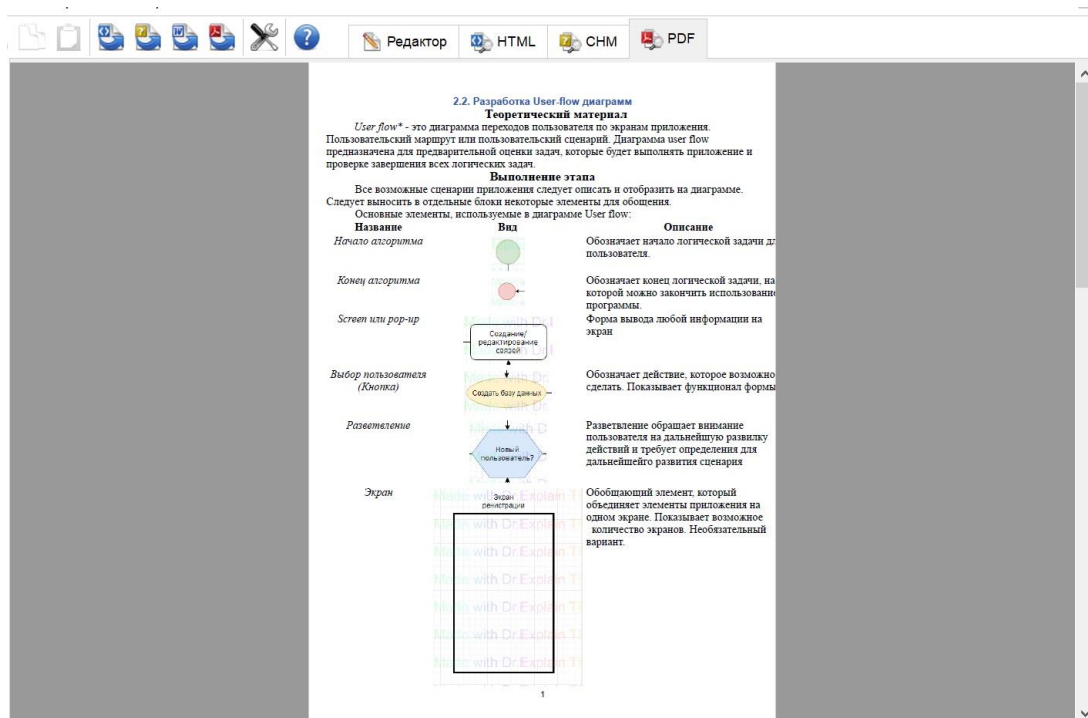


Рис. 6. Редактор pdf-документов.

- ✓ Быстрое описание структуры пользовательского интерфейса. При добавлении в электронный документ скриншота описываемой программы, dr.explain выделяет все компоненты и добавляет к каждому сноску для детализированного описании информации;
- ✓ Редактирование аннотаций изображений;
- ✓ Добавление к каждому разделу оглавления статусов реализации. Для всех разделов доступны статусы: “не начато”, “в процессе”, “закончено” и “ожидается”.

Недостатками платформы для разработки электронных документов является ограниченная платная версия продукта. Хотя бесплатная версия является полноценным продуктом для разработки электронных документов, при приобретении платной версии станут доступны функции добавления изображений без водяных символов компании и онлайн редактирование одного документа с



другими пользователями сети. Онлайн редактирование предоставляется с помощью облачной платформы twiri.com.

Таким образом, на основе проведенного анализа программных продуктов, для разработки руководства пользователя «Технология проектирования адаптивного дизайна» была выбрана программа «Dr. Explain», так как имеет большое количество преимуществ и русскую локализацию программного интерфейса.

### **1.3 Техническое задание на разработку информационной системы (технологии, программного продукта и пр.) «Технология проектирования адаптивного дизайна»**

#### **1. Общие сведения.**

##### **1.1. Название организации-заказчика.**

Уральский Государственный Педагогический Университет кафедра информационно-коммуникационных технологий в образовании.

##### **1.2. Название продукта разработки (проектирования).**

Руководство пользователя “Технология проектирования адаптивного дизайна”

##### **1.3. Назначение продукта.**

Освоение технологии проектирования адаптивного дизайна.

##### **1.4. Плановые сроки начала и окончания работ.**

В соответствии с графиком учебного процесса с 10.01.2019 по 20.03.2019.

#### **2. Характеристика области применения продукта.**

2.1. Процессы и структуры, в которых предполагается использование продукта разработки.

Проектирование и разработка программного продукта для мобильного устройства под управлением операционной системы Android.

2.2. Характеристика персонала (количество, квалификация, степень готовности)

2.2.1. Специалист по подготовке электронных документов – знание пакета Microsoft Office, ГОСТов для создания электронных документов. Опыт работы с электронными документами не менее 1 года.

2.2.2. Дизайнер – знание Adobe Illustrator или другого векторного редактора, Adobe Photoshop или другого растрового редактора. Опыт разработки дизайна для веб-сайтов или мобильных приложений.

### 3. Требования к продукту разработки.

#### 3.1. Требования к продукту в целом.

Требование к руководству пользователя “Технология проектирования адаптивного дизайна”: возможность контекстного поиска, наличие подробных иллюстраций, поэтапное описание разработки адаптивного дизайна для мобильных приложений

#### 3.2. Аппаратные требования.

ПК с оперативной памятью 1 ГБ (для 32-разрядных систем) и 2 ГБ (для 64-разрядных систем), 3 ГБ свободного места на диске, для использования аппаратного ускорения требуется видеоадаптер, поддерживающий DirectX 10. Смартфон с поддержкой стандарта передачи данных не ниже, чем 3G для доступа в сеть интернет.

3.3. Указание системного программного обеспечения (операционные системы, браузеры, программные платформы и т.п.).

Для смартфонов операционная система Android не ниже версии 7.0. Для ПК операционная система Windows 7.

3.4. Указание программного обеспечения, используемого для реализации.

Программное обеспечение для создания руководства пользователя - Dr. Explane. Программное обеспечение для просмотра руководства пользователя - браузер для просмотра формата \*.HTML, программа для чтения электронных книг используется для просмотра формата \*.CHS, текстовый процессор для

просмотра формата\*.RTF, инструмент для просмотра документов в формате \*PDF.

Программное обеспечение для разработки графических объектов - Adobe Illustrator или любой другой векторный редактор

### 3.5. Форматы входных и выходных данных

Текстовый поисковый запрос по руководству пользователя.

3.6. Источники данных и порядок их ввода в систему (программу), порядок вывода, хранения

Текстовый поисковый запрос по руководству пользователя.

3.7. Порядок взаимодействия с другими системами, возможности обмена информацией.

Не предусмотрено.

### 3.8. Меры защиты информации.

Авторские права.

## 4. Требования к пользовательскому интерфейсу.

### 4.1. Общая характеристика пользовательского интерфейса.

Стандартная справка HTML. Документы, оформленные по ГОСТу, задающие структуру содержания.

### 4.2. Размещение информации на экране, дизайн экрана.

Дизайн экрана и размещение элементов должно соответствовать представленным эскизам.

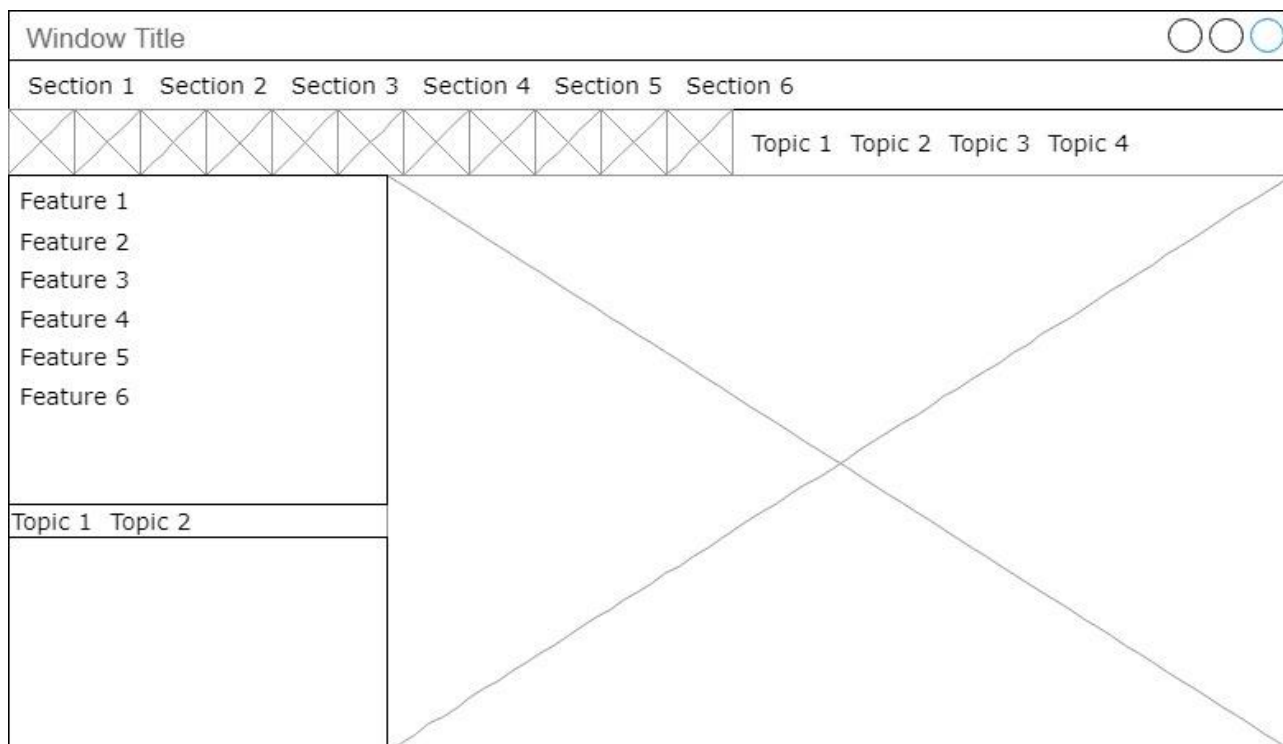


Рис. 7. Эскиз рабочей области

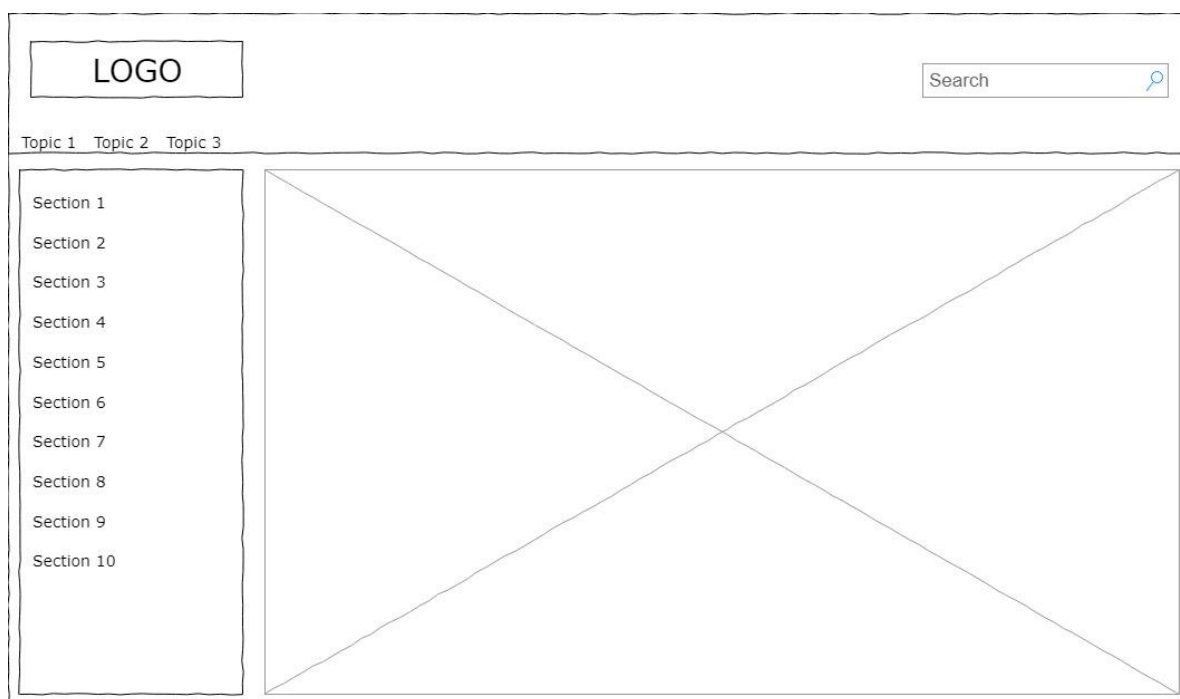


Рис. 8. Эскиз html-документа



Рис. 9. Эскиз chs-документа.

4.3. Особенности ввода информации пользователем, представление выходных данных.

Руководство пользователя представляется в форматах \*.html, \*.pdf, \*.chm, \*.rtf.

5. Порядок сдачи-приемки продукта.

В соответствии с графиком выполнения выпускной квалификационной работы [15, 16].

## Глава 2. Реализация адаптивного дизайна по технологии Material Design представленного в формате руководства пользователя

### 2.1 Структура руководства пользователя по реализации адаптивного дизайна по технологии Material Design

Для реализации технологии адаптивного дизайна по технологии Material Design представим структуру руководства пользователя, задающую последовательность разработки руководства для разработчика и алгоритм освоения технологии для пользователя.

В ходе работы были разработаны функциональные диаграммы IDEF0. Рассмотрим их более подробно. На уровне A0 (Рис. 10) функциональный блок «Технология проектирования адаптивного дизайна» принимает на вход какой-либо поисковый запрос в текстовом формате, в качестве управляющих объектов выступают ГОСТы, механизмами пользователя – является пользователь руководства, на выходе формируется (открывается статья) справочная информация по выбору пользователя.

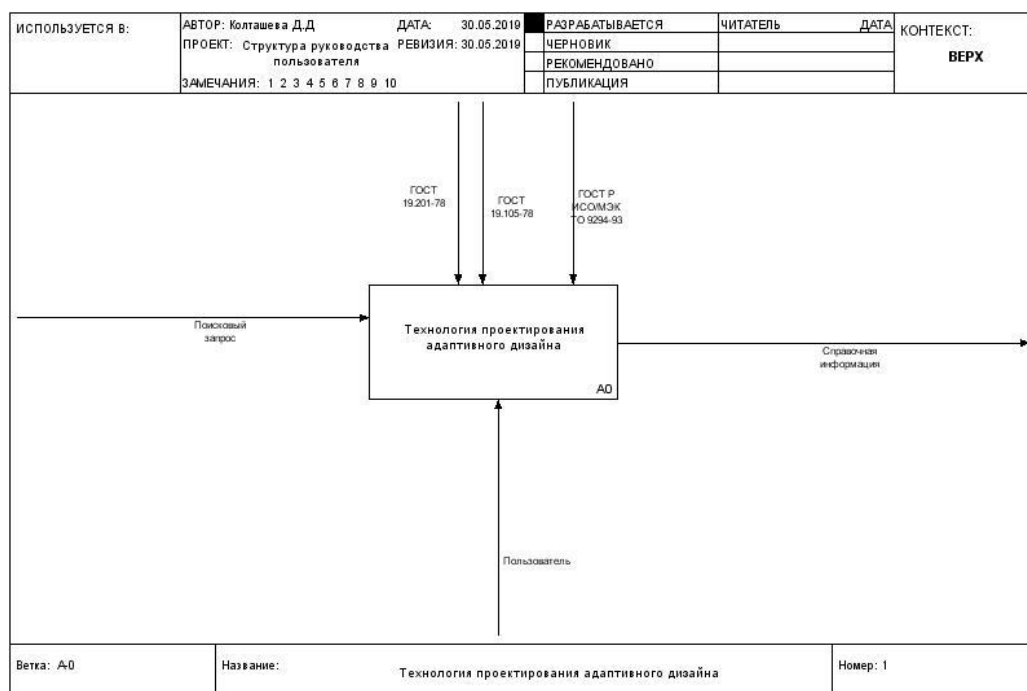


Рис. 10. Функциональный блок «Технология проектирования адаптивного дизайна»

На диаграмме декомпозиции блока А0 (Рис. 11) имеется 3 функциональных блока:

- введение;
- этапы разработки дизайна;
- полезные материалы.

Функциональный блок «Введение» содержит в себе:

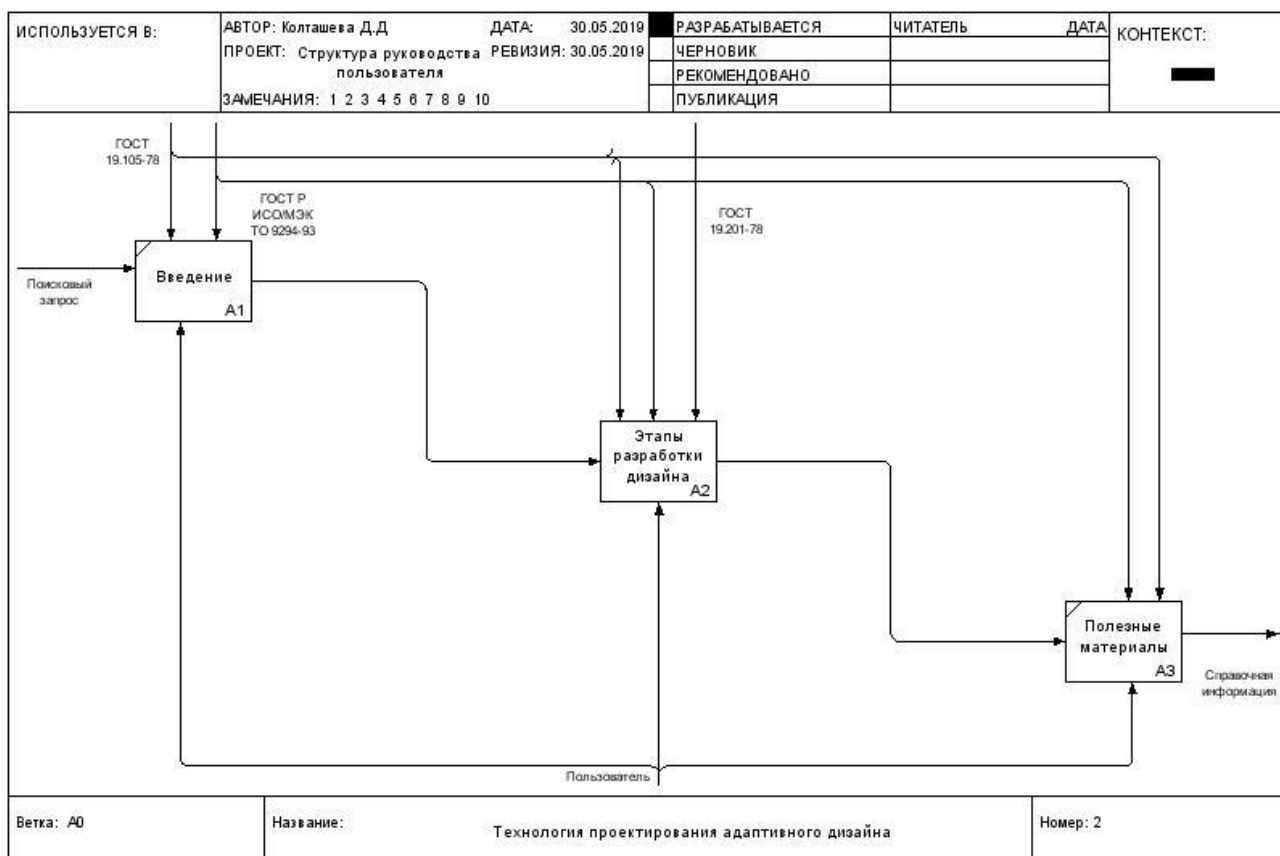
- вход: поисковый запрос;
- управление: ГОСТ 19.105-78 и ГОСТ Р ИСОМЭК ТО 9294-93;
- механизм: пользователь;
- выход: поисковый справочная информация

Функциональный блок «Этапы разработки дизайна» содержит в себе:

- вход: поисковый запрос;
- управление: ГОСТ 19.105-78, ГОСТ 19.201-78 и ГОСТ Р ИСОМЭК ТО 9294-93;
- механизм: пользователь;
- выход: поисковый справочная информация

Функциональный блок «Полезные материалы» содержит в себе:

- вход: поисковый запрос;
- управление: ГОСТ 19.105-78 и ГОСТ Р ИСОМЭК ТО 9294-93;
- механизм: пользователь;
- выход: поисковый справочная информация



*Рис. 11. Декомпозиция первого уровня функционального блока*

На диаграмме декомпозиции блока A2 (Рис. 12) имеется 7 функциональных блоков, представляющих собой последовательную схему разработки дизайна:

- «Изучение требований заказчика к проекту»;
- «Разработка user-flow диаграмм»;
- «Разработка эскизов для всех экранов»;
- «Определение паттернов и цветовой палитры»;
- «Разработка адаптивных прототипов экрана»;
- «Тестирование прототипов пользователями»;
- «Внесение правок и запуск»;

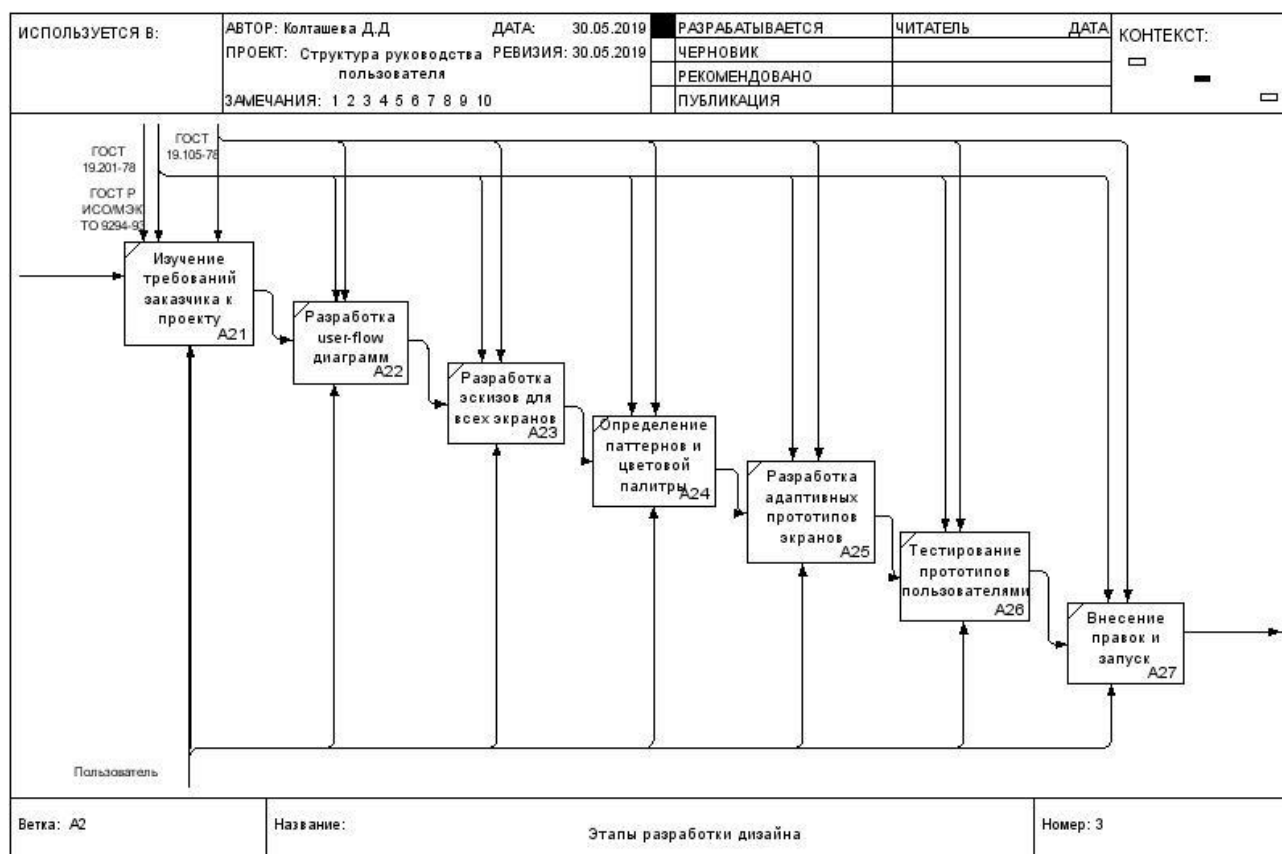
Все перечисленные функциональные блоки содержат в себе:

- вход: поисковый запрос;
- управление: ГОСТ 19.105-78 и ГОСТ 19.201-78;
- механизм: пользователь;



- выход: поисковый справочная информация

Функциональный блок «Изучение требований заказчика к проекту» содержит в себе дополнительную стрелку управления ГОСТ Р ИСОМЭК ТО 9294-93.



*Рис. 12. Декомпозиция второго уровня функционального блока*

На структурной диаграмме отражена структура этапа «разработка адаптивных прототипов экранов», она включает в себя такие блоки как: краткий теоретический материал, описание пунктов реализации этапа, описание программного обеспечения, в котором реализуется данный этап и т.д. (Рис. 13)



*Рис. 13. Структура одного из этапов на примере этапа «разработка адаптивных прототипов экранов»*

На основе спроектированных функциональных диаграмм разработано руководство пользователя по реализации технологии «Технология проектирования адаптивного дизайна».

## **2.2 Руководство пользователя «Технология проектирования адаптивного дизайна»**

В соответствии с ГОСТ 34.602-89 структурные элементы руководства представлены далее в формате содержания [20, 22]:

1. Введение
  - 1.1 Область применения
  - 1.2 Краткое описание возможностей
  - 1.3 Перечень эксплуатационной документации, с которыми необходимо ознакомиться пользователю
2. Назначение и условия применения
  - 2.1 Виды деятельности, функции, для автоматизации которых предназначено данное средство автоматизации
3. Подготовка к работе

### 3.1 Порядок загрузки данных и программ

#### 4. Рекомендации к освоению

1. Введение
2. Этапы разработки дизайна
  - a. Изучение требований заказчика к проекту
  - b. Разработка user-flow диаграмм
  - c. Разработка эскизов для всех экранов
  - d. Определение паттернов и цветовой палитры.
  - e. Разработка адаптивных прототипов экранов.
  - f. Тестирование прототипов пользователями.
  - g. Внесение правок и запуск.
3. Полезные материалы

## 1. Введение

В руководстве пользователя "Технология разработки адаптивного дизайна" представлена технология разработки многослойного дизайна приложения под ОС Android. Иллюстративный материал подготовлен на примере приложения "EasyBox".

Дизайн мобильного приложения играет важную роль в разработке всего мобильного приложения, поскольку является одним из критериев, по которому пользователь решает, будет ли он использовать данный программный продукт.

Нравится ли внешний вид приложения пользователю? Удобно ли его использовать? Понятно ли ему для каких целей предназначено ваше приложение? Привычно ли расположены элементы взаимодействия? На все эти вопросы отвечает дизайн интерфейса приложения.

Для полноценного изучения разработки адаптивного дизайна, следует рассмотреть основные этапы создания дизайна для мобильных приложений, такие как:

- Изучение требований заказчика к проекту (ТЗ).

- Разработка user-flow диаграмм для всех экранов на основе требований заказчика.
- Разработка эскизов для всех экранов приложения.
- Определение паттернов и цветовой палитры.
- Разработка адаптивных прототипов экранов.
- Тестирование прототипов пользователями.
- Внесение правок и запуск.

Для понятий, отмеченных символом "\*", в конце каждой главы в разделе Словарь представлены формулировки их определений.

## 2. Этапы разработки дизайна

В каждом из этапов разработки дизайна мобильного приложения будут рассмотрены:

- краткий теоретический материал;
- описание пунктов реализации этапа;
- описание программного обеспечения, в котором реализуется данный этап;
- готовый пример;
- словарь с основными понятиями;

### 2.1. Изучение требований заказчика к проекту

#### *Теоретический материал*

В начале разработки любого мобильного приложения, дизайнер, вместе с заказчиком, должен создать *техническое задание\**, в котором будут рассмотрены основные требования к внешнему виду приложения. ТЗ может быть создано по *ГОСТу\** к определенному продукту или оформлено в виде неформального документа.

*Пример технического задания для приложения "EasyBox".*

1. Требования к пользовательскому интерфейсу.
1. Общая характеристика пользовательского интерфейса.

Высококонтрастное цветовое решение на основе холодных тонов для комфортного использования приложения при естественном освещении. Допускается использование цветовой палитры стилистики Material Design.

2. Размещение информации на экране, дизайн экрана.

Для комфортной работы с мобильным приложением использовать только книжную ориентацию экрана. Использовать боковое меню для размещения в нем вкладки генерации и настроек. Допускается использование простейших паттернов для реализации интерфейса и стилистики Material Design.

3. Особенности ввода информации пользователем, представление выходных данных.

Подразумевается, что пользователь будет производить ввод информации прикосновением по экрану устройства. Выходные данные представляются в виде окна с редактируемым текстом, в котором находится SQL код.

### *Словарь [18]*

*Техническое задание* (часто используется сокращение ТЗ) - специальный документ, который разработан заказчиком и исполнителем, в котором сформулированы основные требования, параметры и характеристики, разрабатываемого продукта.

*ГОСТ* — это государственный стандарт, который формулирует требования государства к качеству продукции, работ и услуг, имеющих межотраслевое значение.

## **2.2. Разработка User-flow диаграмм**

### *Теоретический материал*

*User-flow\** — это диаграмма переходов пользователя по экранам приложения. Пользовательский маршрут или пользовательский сценарий. Диаграмма


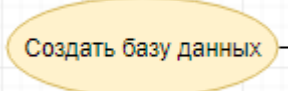
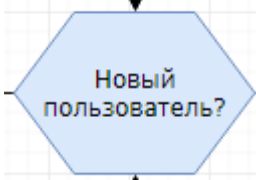
user-flow предназначена для предварительной оценки задач, которые будет выполнять приложение и проверке завершения всех логических задач.

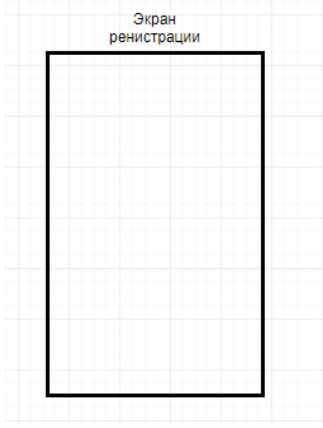
### *Выполнение этапа*

Все возможные сценарии приложения следует описать и отобразить на диаграмме. Следует выносить в отдельные блоки некоторые элементы для обобщения.

Основные элементы, используемые в диаграмме user-flow:

*Таблица 2.*  
*Основные элементы диаграммы user-flow.*

Название	Вид	Описание
Начало алгоритма		Обозначает начало логической задачи для пользователя.
Конец алгоритма		Обозначает конец логической задачи, на которой можно закончить использование программы
Screen или pop-up		Форма вывода любой информации на экран
Выбор пользователя (Кнопка)		Обозначает действие, которое возможно сделать. Показывает функционал формы
Разветвление		Разветвление обращает внимание пользователя на дальнейшую развилку действий и требует определения для дальнейшего

		развития сценария
Экран		Обобщающий элемент, который объединяет элементы приложения на одном экране. Показывает возможное количество экранов. Необязательный вариант

Придерживайтесь некоторых правил, по созданию диаграммы:

- Стрелки, соединяющие элементы, не должны проходить под элементами. Стрелка должна быть полностью видна.
- Рекомендуется к использованию фигур со скругленными углами для того, чтобы стрелка не пересеклась с углом и её можно было бы легко опознать. Также такие фигуры помогут распознавать объекты друг от друга, которые находятся на стыках линий.
- Стрелки могут объединяться в одну, когда они выполняют равнозначные функции.
- Приветствуется использование комментариев и подписей для уточнений.

#### *Программное обеспечение*

Разработка диаграмм возможна в любом графическом редакторе.

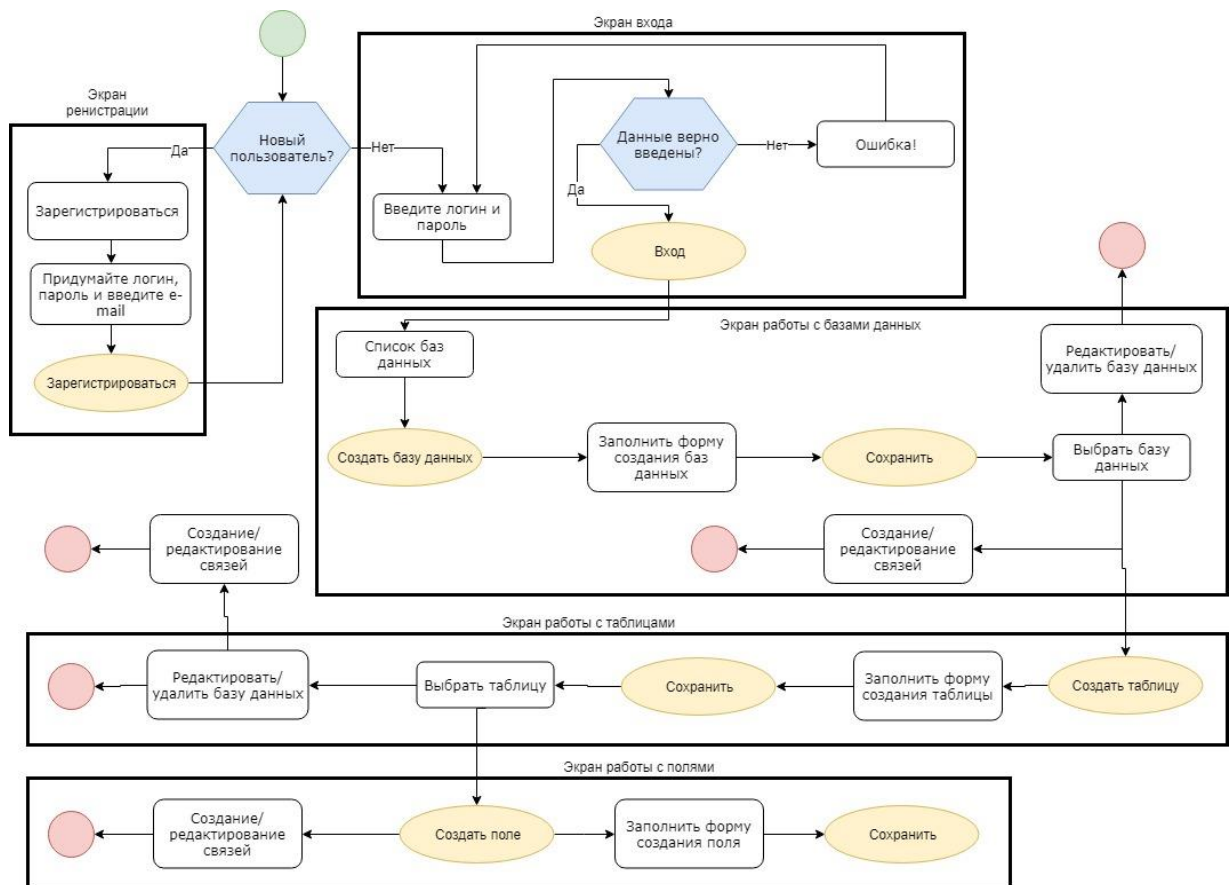
Например:

- Векторный редактор:
  - Adobe Illustrator;
  - Corel Draw;
  - Подобные
- Растровый редактор:
  - Adobe Photoshop;

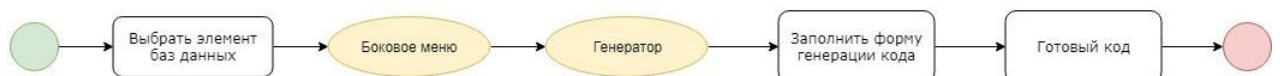
- Gimp
- Подобные
- Онлайн-редактор:
  - draw.io
  - gliffy
  - Подобные

Рекомендуется использовать онлайн-редакторы, потому что в них есть готовые примеры и коллекции геометрических фигур, используемых при создании диаграммы.

*Пример готовой диаграммы.*



*Рис. 14. User flow диаграмма.*



*Рис. 15. User flow диаграмма.*



## Словарь

*User experience flow (user-flow)* — переход пользователей от одного сценария взаимодействия с интерфейсом к другому, пользовательский маршрут.

### 2.3. Разработка эскизов для всех экранов

#### *Теоретический материал*

После того как основные цели и задачи мобильного приложения сформулированы, и разработан пользовательский маршрут приложения, можно приступать к созданию примерных эскизов приложения.

При создании эскизов необходимо учитывать принципы *UX/UI\** дизайна. Основные принципы:

- простой *интерфейс\** - задачи, решаемые пользователем, должны решаться минимальным количеством действий. Отсутствие сложных логических цепочек для достижения целей.
- отсутствие очевидных элементов - если стоит выбор между размещением элемента или не размещением, лучше не добавлять такой элемент, чтобы не нагружать экран приложения
- использовать привычное размещение и привычные элементы — это позволит пользователю быстрее сориентироваться в вашем интерфейсе.
- избегать большого количества текста - пользователи не любят громоздкий и длинный текст. Текст, по возможности, необходимо сокращать до минимума. Если это невозможно, текст принято разделять по маленьким абзацам или размещать в таблице.
- короткие функциональные блоки - исходя из исследования Миллера, человек может удерживать в кратковременной памяти не более 5-7 элементов.
- использование принципа трех кликов - принцип трех кликов подразумевает, что для достижения цели, пользователю необходимо сделать не более трех кликов, иначе это будет слишком длинный путь до цели. Все важные элементы для достижения цели должны располагаться на главном экране.

- принцип однородности - большинство функционала приложения должно быть однородным. Также, как и однородностью должен обладать стиль приложения, иначе это может дезориентировать пользователя.
- принцип единства - основные элементы и настройки приложения должны по возможности располагаться в одном месте.

### *Выполнение этапа*

Для начала необходимо определиться с количеством экранов приложения. Количество считается по задачам, которые выполняют приложения. После определения количества экранов приложения необходимо продумать функционал каждого экрана и основные элементы. Сгруппировать однотипные экраны, как например экран создания какого-либо материала, скорее всего, будет выглядеть как привычная пользователям форма. В форме можно использовать текстовые поля, выпадающие списки, флажки и прочее.

### *Пример*

Приложение "EasyBox" предназначено для генерации SQL-кода на основе спроектированной базы данных. Для начала работы с приложением необходимо разработать базу данных, которая состоит из таблиц, которые в свою очередь состоят из полей. Также поля таблиц можно связывать друг с другом, из этого следует, что связи нужно тоже создавать. Таким образом, получается, что для каждого из элементов базы данных необходимы экраны создания, представления, редактирования. Далее необходим экран создания и представления SQL-кода. Также в этом приложении должен быть реализован вход и регистрация. Экраны можно разбить на следующие типовые группы:

- Вход и регистрация;
- Экраны создания (Создание базы данных, таблиц, полей и связей);
- Экраны представления (Представление существующих баз данных, таблиц, полей и связей);
- Боковое меню (Для каждого элемента баз данных);

- Панель редактирования (Для каждого элемента баз данных);
- Генерация кода (Создание и представление кода);

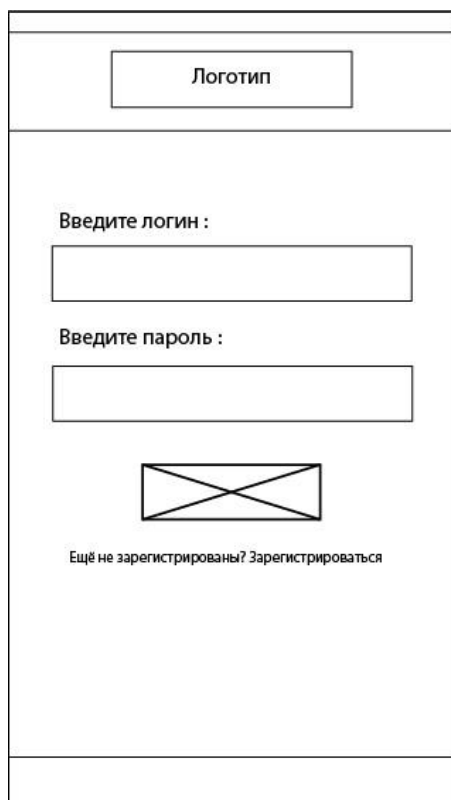
Таким образом всего в приложении 20 экранов.

Руководствуясь техническим заданием, необходимо разработать боковое меню и поместить его на все экраны приложения.

Разработка диаграмм возможна в любом графическом редакторе. Например:

- Векторный редактор:
  - Adobe Illustrator;
  - Corel Draw;
  - Подобные
- Растровый редактор:
  - Adobe Photoshop;
  - Gimp
  - Подобные
- Онлайн-редактор:
  - draw.io
  - gliffy
  - Подобные

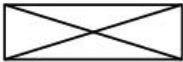
*Примеры готовых эскизов.*



Логотип

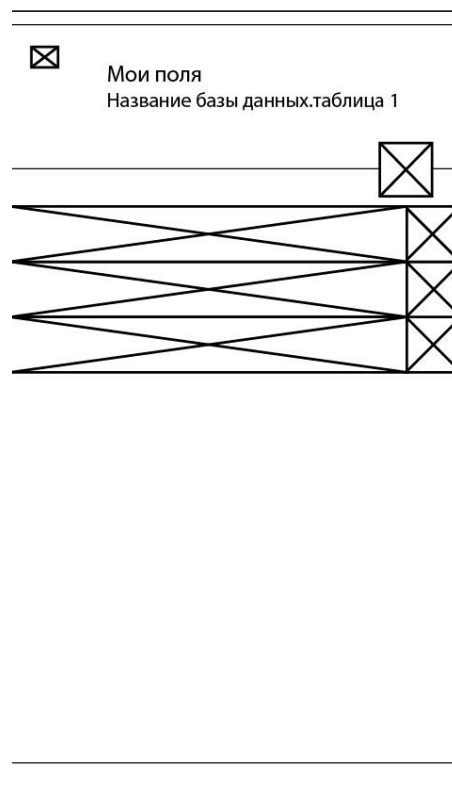
Введите логин :


Введите пароль :



Ещё не зарегистрированы? Зарегистрироваться

Рис. 16. Вход



 Мои поля  
Название базы данных.таблица 1


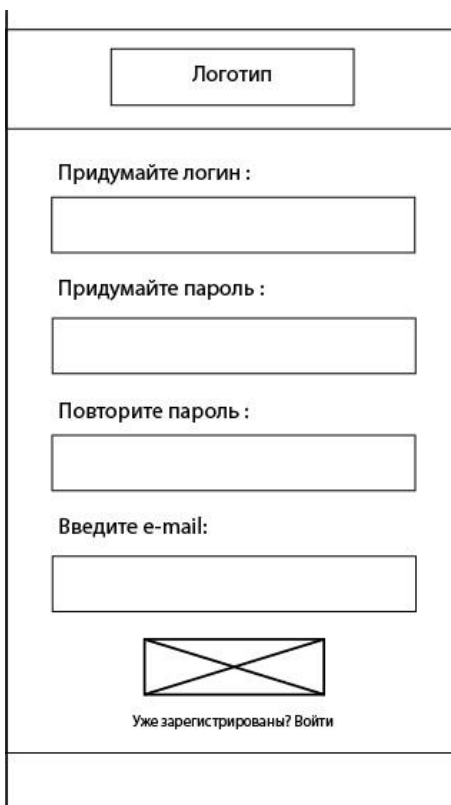



Рис. 17. Мои поля




Логотип

Придумайте логин :

Придумайте пароль :

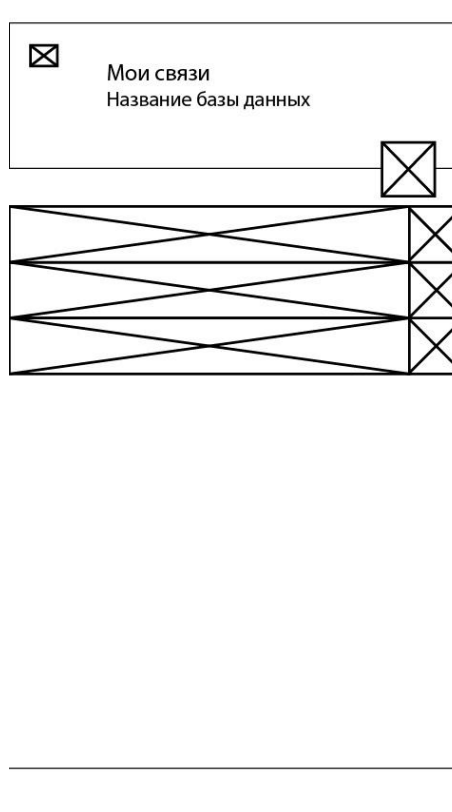
Повторите пароль :


Введите e-mail:



Уже зарегистрированы? Войти

Рис. 18. Регистрация



 Мои связи  
Название базы данных





Рис. 19. Моя связи



Название бд1

		X
		X
		X

Название

Первичный ключ

Внешний ключ

Тип

При удалении

При обновлении

Сохранить

Рис. 24. Создание связи

Название бд 1

		X
		X
		X

Название

Сохранить

Рис. 25. Создание таблицы

username e-mail

Главная

Настройки

Выход

		X
		X
		X

Рис. 26. Боковое меню базы данных

username e-mail

Главная

Настройки

Выход

		X
		X
		X

Рис. 27. Боковое меню поля

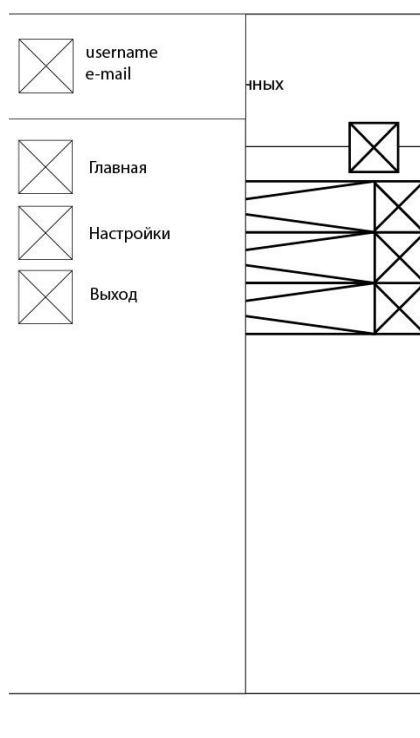


Рис. 28. Боковое меню связи

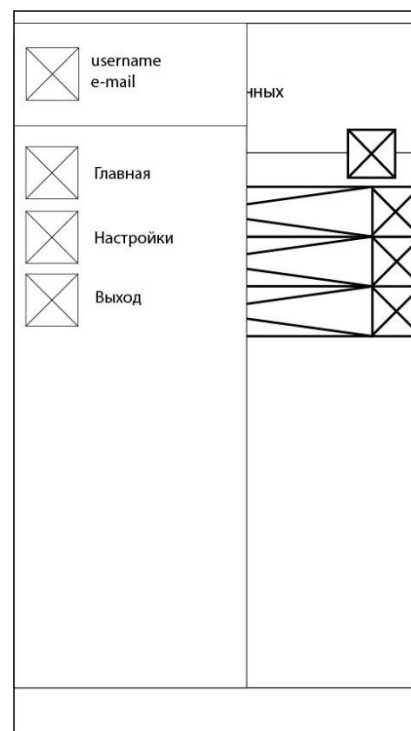


Рис. 29. Боковое меню таблицы



Рис. 30. Контекстное меню базы данных

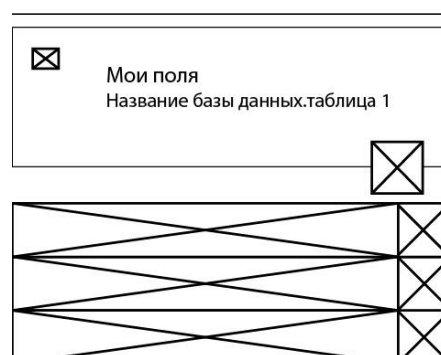


Рис. 31. Контекстное меню поля

<input checked="" type="checkbox"/> Мои связи Название базы данных	
<input checked="" type="checkbox"/>	
<div style="display: flex; justify-content: space-between;"> <div style="width: 80%;"> </div> <div style="width: 15%;"> <input checked="" type="checkbox"/> </div> </div>	
<div style="display: flex; justify-content: space-between;"> <div style="width: 80%;"> </div> <div style="width: 15%;"> <input checked="" type="checkbox"/> </div> </div>	
<div style="display: flex; justify-content: space-between;"> <div style="width: 80%;"> </div> <div style="width: 15%;"> <input checked="" type="checkbox"/> </div> </div>	
<div style="display: flex; justify-content: space-between;"> <div style="width: 80%;"> </div> <div style="width: 15%;"> <input checked="" type="checkbox"/> </div> </div>	
<input checked="" type="checkbox"/> Изменить	
<input checked="" type="checkbox"/> Удалить	
<input checked="" type="checkbox"/> Связь	

Рис. 32. Контекстное меню связи

<input checked="" type="checkbox"/> Мои таблицы Название базы данных	
<input checked="" type="checkbox"/>	
<div style="display: flex; justify-content: space-between;"> <div style="width: 80%;"> </div> <div style="width: 15%;"> <input checked="" type="checkbox"/> </div> </div>	
<div style="display: flex; justify-content: space-between;"> <div style="width: 80%;"> </div> <div style="width: 15%;"> <input checked="" type="checkbox"/> </div> </div>	
<div style="display: flex; justify-content: space-between;"> <div style="width: 80%;"> </div> <div style="width: 15%;"> <input checked="" type="checkbox"/> </div> </div>	
<div style="display: flex; justify-content: space-between;"> <div style="width: 80%;"> </div> <div style="width: 15%;"> <input checked="" type="checkbox"/> </div> </div>	
<input checked="" type="checkbox"/> Изменить	
<input checked="" type="checkbox"/> Удалить	
<input checked="" type="checkbox"/> Связь	

Рис. 33. Контекстное меню таблицы

<input checked="" type="checkbox"/> Генератор	
Выберите базу данных	<input type="text"/>
Выберите тип базы данных	<input type="text"/>
<div style="display: flex; justify-content: space-between;"> <div style="width: 80%;"> </div> <div style="width: 15%;"> <input checked="" type="checkbox"/> </div> </div>	

Рис. 34. Генерация кода.

<input checked="" type="checkbox"/> Генератор	
Выберите базу данных	<input type="text"/>
Выберите тип базы данных	<input type="text"/>
<div style="display: flex; justify-content: space-between;"> <div style="width: 80%;"> </div> <div style="width: 15%;"> <input checked="" type="checkbox"/> </div> </div>	
Сгенерированный код Сгенерированный код Сгенерированный код	

Рис. 35. Генерация кода и код.



## *Словарь*

*UX (user experience)* - дословно: пользовательский опыт или опыт взаимодействия [27]. Это восприятие и ответные действия пользователя, возникающие в результате использования какого-либо программного продукта либо предстоящего использования. Кратко: эмоции или ощущения после, или перед использованием программного продукта или услуги/сервиса.

*UI (user interface)* - дословно: пользовательский интерфейс или интерфейс пользователя. Это внешний вид, дизайн интерфейса всего приложения. Как будет выглядеть та или иная кнопка, какого цвета будет элемент, и прочие вопросы решаются с помощью UI.

*интерфейс* - совокупность средств, методов и правил взаимодействия (управления, контроля и т. д.) между элементами системы.

## **2.4. Определение паттернов и цветовой палитры**

### *Теоретический материал*

Паттерны имеют различные значения, в данном контексте паттерн рассматривается, как готовый шаблон мобильного приложения, который можно использовать в качестве примера [4]. Цветовые схемы играют важную роль в создании дизайна приложения. Если цвета не гармонируют между собой или плохо контрастируют цвета фона и текста, то пользователь, скорее всего, откажется от использования такого приложения.

### *Выполнение этапа*

Выбор паттерна является примером как может выглядеть ваше приложение, с какими элементами, где и в каких цветах. После примерного определения паттерна проще реализовывать эскизы и готовый дизайн. Цветовые схемы определяются в соответствии со стилистикой приложения и спецификацией.

### *Пример*

Приложение "EasyBox" предназначено для генерации SQL-кода на основе спроектированной базы данных. Данное приложение будет разработано в стиле Material Design, так как использование данной стилистики допускается в техни-

ческом задании. Рассмотрим палитру цветов стилистики Material Design [7]. На сайте "materialpalette.com" интерактивно представлено изображения простейшего паттерна и цветовой палитры, в которой вы сами выбираете 2 основных цвета, а все остальное представление будет выполнено за вас (Рис. 36).

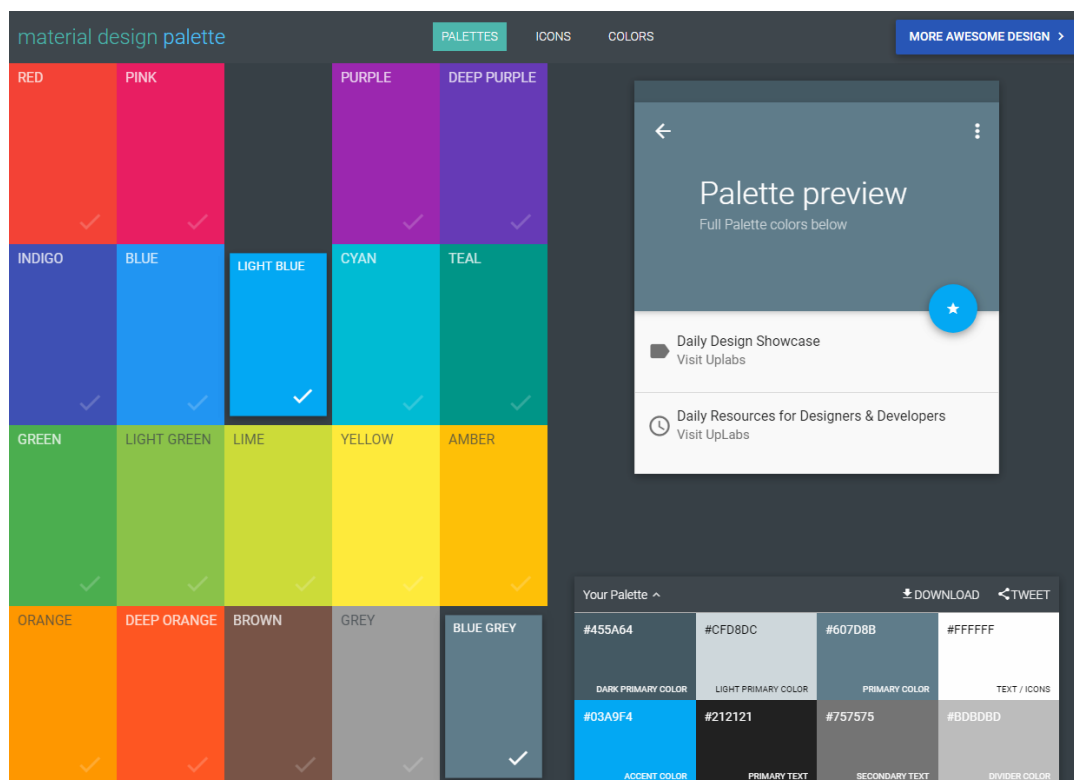


Рис. 36. Выбор цветовой палитры и паттерна

В данном случае выбирается основной цвет приложения и акцентный цвет, на основе, заданных вами цветов, подберутся остальные оттенки для приложения [6]. Выбор цветовой палитры для мобильного приложения «EasyBox» должен соответствовать техническому заданию. В техническом задании рекомендуется использовать холодные тона и высококонтрастные сочетания оттенков. Основываясь на ТЗ нам подходит следующая палитра (Рис. 37).

*Пример выбранной палитры.*



*Рис. 37. Цветовая палитра.*

Назначение цветов, соответствующих палитре.

*Таблица 3.*

*Соответствие назначений цветов и палитре цветов.*

Темный основной цвет	Светлый основной цвет	Основной цвет	Фоновый цвет
Акцентный цвет	Цвет для основного текста	Цвет для второстепенного текста	Цвет разделителя

### *Словарь*

*Паттерн* - готовый шаблон приложения, предназначенный для использования в качестве примера.

## **2.5. Разработка адаптивных прототипов экранов**

### *Теоретический материал*

Прототип — это окончательный вид приложения, разрабатываемый на основе эскизов, цветовых палитр и паттернов, которые вы определили для приложения. Адаптивный прототип подразумевает разработку нескольких прототипов, определенных размеров для того, чтобы на разных диагоналях различных устройств, приложение выглядело единообразно.

Для создания прототипов в едином стиле необходимо определиться со стилистикой будущего приложения. Рассмотрим самую популярную стилистику для мобильных приложений под управление ОС Android – Material Design.

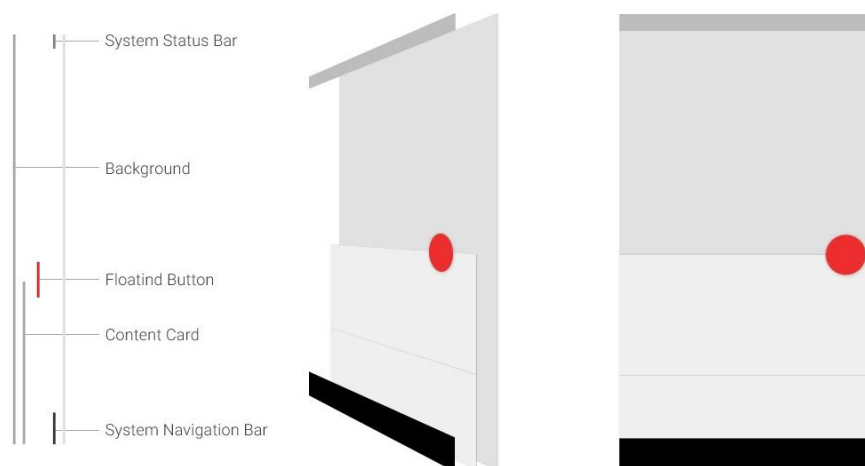
Material Design – это специальная стилистика мобильных приложений разработанная компанией Google в 2014 году [1]. Основная идея материального

дизайна – многослойность. Экран устройства можно рассмотреть не только в двухмерной плоскости, но и в трехмерной плоскости. Этот подход дает возможность располагать элементы друг над другом, за счет использования теней у каждого элемента [5]. Очевидно, что самым первым элементом, на котором будут расположены все элементы является фон, поэтому он не должен иметь тени, но все оставшиеся элементы должны иметь тень, причем чем «дальше» от фона находится элемент, тем большую тень он должен отбрасывать. Многослойность дизайна необходима для того, чтобы показать пользователю основные правила и принципы работы программного обеспечения, переходы состояний и т.д.

Анимации в материал дизайн также носят не только развлекательный или отвлекающий, но и информационный характер. Так множество анимаций могут объяснить пользователю, что делает тот или иной объект или показывать некоторые подсказки для продолжения корректной работы пользователя с программным обеспечением.

Материал дизайн был разработан для унификации интерфейсов мобильных приложений под управлением операционной системы Android [7,8]. Материальный дизайн использует в своей разработке 4 основных принципа:

1. **Тактильные поверхности** – реализует многослойность. Объясняет «глубину», тени и трёхмерность (Рис. 38).
2. **Полиграфический дизайн** – использует традиционные стили типографики журнальный или плакатный.
3. **Осмысленная анимация** – работает для пользователя как информационные правила
4. **Адаптивный дизайн** – унифицирует дизайн приложения для экранов устройств с разными разрешениями и размерами.



*Рис. 38. Пример многослойности экрана*

Среди мобильных устройств, под управлением ОС Android, множество с разными диагоналями экранов. Главная задача дизайнера мобильных приложений разработать такой дизайн для приложения, который бы смотрелся на всех устройствах единообразно. Для решения этой задачи необходимо знать, что такое плотность экрана\* и какие размеры прототипов необходимо задать.

Плотность экрана — это показатель пикселей на один дюйм экрана, другими словами, если взять "кусочек" экрана размером 2,54 см (1 дюйм = 2,54 см), то сколько в этом "кусочке" будет пикселей. Далее представлена таблица различных плотностей экранов и размеров.

*Таблица 4.  
Зависимость плотности экрана от размера экрана.*

Плотность	Множитель	Размер экрана
ldpi	0.75x	240*360
mdpi (стандарт)	1.0x	320*480
hdpi	1.5x	480*720
xhdpi	2.0x	640*960
xxhdpi	3.0x	960*1280
xxxhdpi	4.0x	1280*1920

Минимальный список файлов, которые дизайнер должен представить разработчику:

- Прототип размером 320\*480, с плотностью экрана mdpi (160 dpi)
- Прототип размером 480\*720, с плотностью экрана hdpi (240 dpi)

- Прототип размером 640\*960, с плотностью экрана xhdpi (320 dpi)
- Прототип размером 960\*1280, с плотностью экрана xxhdpi (320 dpi)

### *Выполнение этапа*

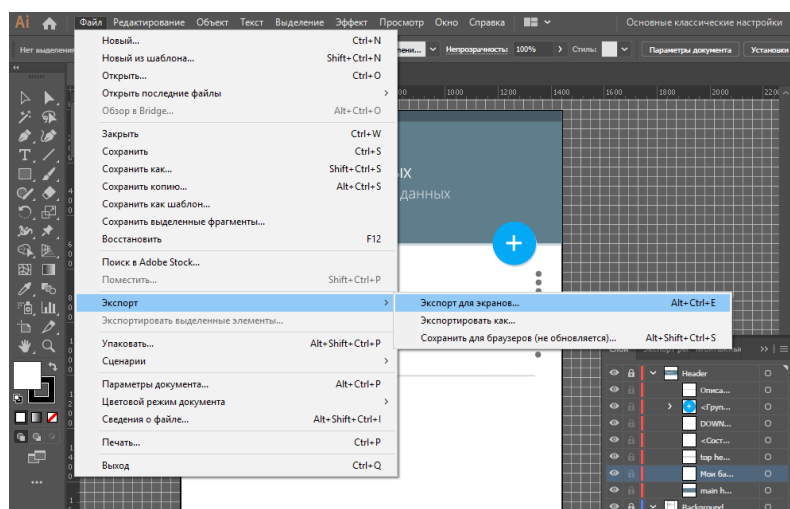
Перед разработкой прототипов экрана необходимо определиться со стилистикой будущего дизайна. В техническом задании на разработку дизайна приложения «EasyBox» указано использование стилистики Material Design.

Для того, чтобы разработанные прототипы смотрелись едино на всех устройствах, необходимо разработать прототип со стандартной плотностью экрана (mdpi). Далее, в зависимости от того, в каком графическом редакторе вы работаете необходимо создать прототипы еще для 3 основных форматов плотности экранов. Создание подразумевает, либо увеличение стандартного прототипа на заданный множитель каждого их форматов плотности, либо экспорт для дисплеев Android.

### *Пример*

Прототипы для приложения "EasyBox" разрабатываются в векторном графическом редакторе Adobe Illustrator, в котором реализован экспорт во все возможные плотности экранов. После того как прототип разработан, необходимо:

1. Выбрать вкладку "Файл" -> "Экспорт" -> "Экспорт для экранов..."(Рис. 39)



*Рис. 39. Скриншот экспорта прототипов.*

2. В появившемся окне выбрать вкладку монтажные области для экспорта всех элементов прототипа. Далее можно выбрать путь, по которому будут созданы папки с фалами. Ниже, в разделе "Форматы" необходимо выбрать вкладку "Android". По умолчанию, для Android, будут созданы все форматы плотности прототипа, но вы можете изменить это по необходимости. Если выбраны все нужные параметры, то нажмите кнопку "Экспорт монтажной области"(Рис. 40).

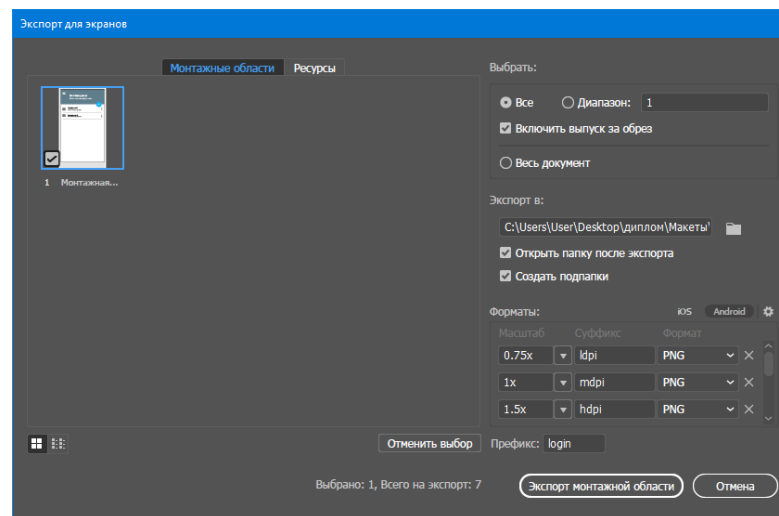


Рис. 40. Скриншот настроек экспорта прототипов.

3. После этого откроется папка, в которой созданы все прототипы с разными плотностями экранов (Рис. 41).

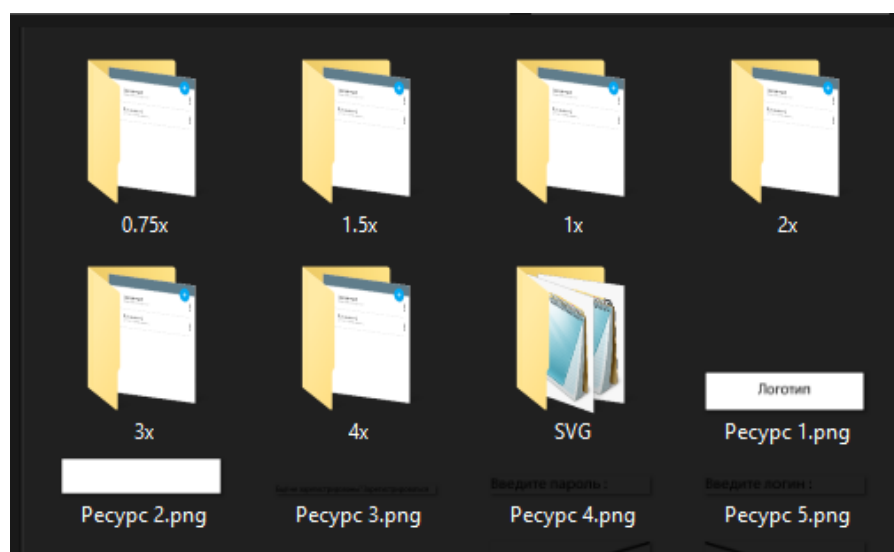


Рис. 41. Экспортированные прототипы экранов.

4. Далее все созданные папки с прототипами и ресурсы, необходимо отправить разработчику приложения для того, чтобы разработчик использовал разработанный вами дизайн далее.

#### *Программное обеспечение*

Разработка диаграмм возможна в любом графическом редакторе.

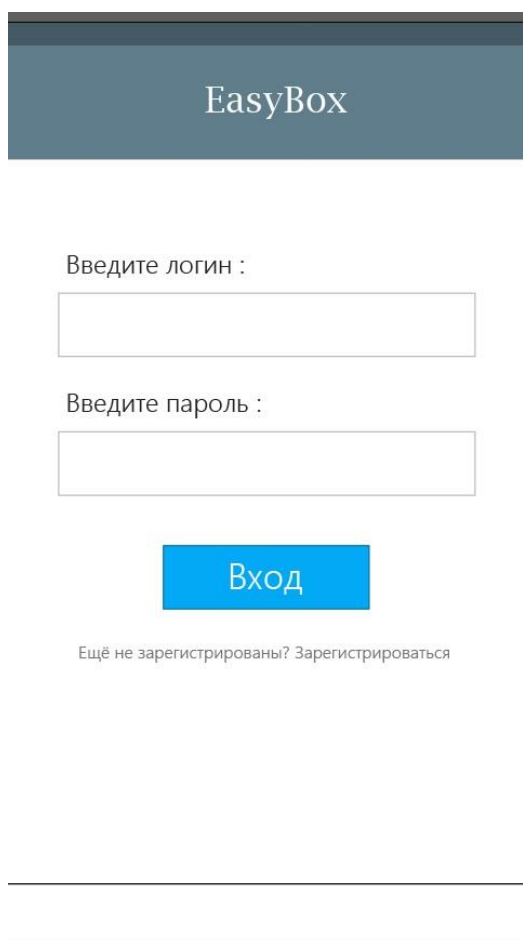
Например:

- Векторный редактор:
  - Adobe Illustrator;
  - Corel Draw;
  - Подобные;
- Растровый редактор:
  - Adobe Photoshop;
  - Gimp;
  - Подобные;
- Онлайн-редактор:
  - draw.io;
  - gliffy;
  - Подобные;

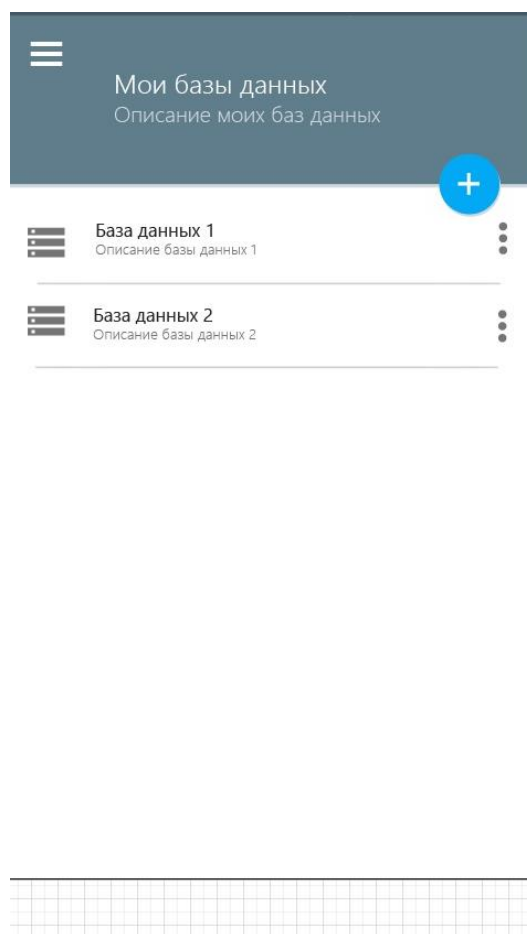
Рекомендуется разрабатывать прототипы дизайна в векторных графических редакторах для возможности качественного масштабирования в дальнейшем.



*Примеры готовых прототипов.*



*Рис. 42. Прототип экрана входа*



*Рис. 43. Прототип главного экрана*

*Словарь*

*Прототип - готовый дизайн*

*Плотность экрана* - иначе называют ppi или pixel per inch - пиксель на дюйм. Показатель количества пикселей на один дюйм экрана устройства.

*Адаптивный дизайн* - дизайн, позволяющий корректно отображать приложения или веб-сервисы на разных устройствах.

## **2.6. Тестирование прототипов пользователями**

*Теоретический материал*

Тестирование разработанных прототипов необходимо для того, чтобы найти все недочеты, которые могли возникнуть при разработке и устранить их до окончательного релиза приложения. Без данного этапа готовое приложение несовершенно. При обнаружении пользователями критических ошибок, сбоев

программы или уязвимостей существует риск потери данных или отказа пользователей от использования приложения.

#### *Выполнение этапа*

Тестирование можно провести различными способами. Например, отправить приложение с дизайном пользователям и узнать отзывы, разработать анкетирование по вашим критериям и отправить для прохождения пользователям, и т.д.

Отзывы после тестирования.

1. «Приложение понравилось. Удобный интерфейс. Критических ошибок не обнаружено.»
2. «Красиво, удобно»
3. «Ошибок не нашел. Нормально»

### **2.7. Внесение правок и запуск готового дизайна.**

#### *Теоретический материал*

После сбора отзывов необходимо приступить к внесению правок по отзывам и запуском готового приложения.

#### *Выполнение этапа*

Отзывы пользователей положительные, критических ошибок не найдено. Приложение может быть запущено.

## **3. Полезные материалы**

1. Список ГОСТов, по которым оформляется ТЗ:
  - а. ГОСТ Р 19.201-78 «Единая система программной документации. Техническое задание. Требования к содержанию и оформлению» от 01.01.1980 № 3351 // Стандартинформ. 2014 г. с изм. и допол. в ред. от 12.09.2018 [15].
  - б. ГОСТ Р 34.602-89 «Информационная технология. Техническое задание на создание автоматизированной системы» от 01.01.1990 // Стандартинформ. 2014 г. с изм. и допол. в ред. от 12.09.2018 [16].

2. Ссылка на МД
  - a. <https://material.io/design/>
  - b. <https://awdee.ru/material-design-full-guide/>
3. Ссылки для поиска паттернов:
  - a. <https://pttrns.com>
  - b. <https://mobile-patterns.com>
4. Ссылка на подбор цветов в МД:
  - a. <https://www.materialpalette.com/colors>

### **2.3 Результаты апробации**

Материалы работы прошли апробацию в формате публикации в сборнике студенческих научных работ. А также на их основе разработан дизайн приложения «EasyBox».

Иванов А.А., Колташёва Д.Д., Сардак Л.В. Методические рекомендации по разработке дизайна в стиле Material design для мобильного приложения под ОС Android // Актуальные вопросы преподавания математики, информатики и информационных технологий [Электронный ресурс] : межвузовский сборник научных работ / Урал. гос. пед. ун-т ; науч. ред. Л. В. Сардак. – Электрон. дан. – Екатеринбург : [б. и.], 2019. – 1 электрон. опт. диск (CD-ROM). – С. 238-241.

## **Заключение**

При разработке мобильных приложений важную роль играют дизайн-эргономические показатели приложения, потому что на их основе пользователь принимает решение об использовании приложения. Перед разработчиками и дизайнерами приложений стоит задача создания масштабируемого и привлекательного интерфейса. С помощью руководства пользователя «Технология проектирования адаптивного дизайна» любой начинающий дизайнер мобильных приложений сможет спроектировать готовые прототипы будущего мобильного приложения.

Цель разработки: создать и описать технологию проектирования адаптивного дизайна в формате руководства пользователя. Продукт разработки – руководство пользователя «Технология проектирования адаптивного дизайна».

Для достижения цели были сформулированы и достигнуты следующие задачи:

1. Произвести анализ информационных источников по теме проектирования интерфейсов для мобильных устройств.
2. Рассмотреть способы представления и описания информационных технологий в справочной форме.
3. В соответствии с техническим заданием провести разработку «Технологии проектирования адаптивного дизайна»
4. Провести апробацию.

Работа носит законченный характер, цель работы достигнута, задачи выполнены. Продукт полностью соответствует техническому заданию и требованиям заказчика.

## Список информационных источников

1. Anatomy // Material Design URL: <https://material.io/design/components/text-fields.html#anatomy> (дата обращения: 21.05.2019).
2. Создание справочной системы или руководства пользователя в Dr.Explain // habr.ru URL: <https://habr.com/ru/company/ivanpr/blog/128876/> (дата обращения: 11.04.2019).
3. Ergogames [Электронный ресурс]: База данных как сервис / Дата обращения: 29.02.2019. — Режим доступа: Intefaces pices
4. Figma URL: <https://www.figma.com/file/Nnc5amC8NjpVjBOr5dTiVMBe/Auditoria?node-id=0%3A1> (дата обращения: 10.04.2019).
5. Icon4android URL: <http://ww11.icons4android.com> (дата обращения: 10.05.2019).
6. Material Design Palette // materialpalette URL: <https://www.materialpalette.com/light-blue/indigo> (дата обращения: 20.05.2019).
7. The color system // Material Design URL: <https://material.io/design/color/the-color-system.html#> (дата обращения: 10.04.2019).
8. Usage // Material Design URL: <https://material.io/design/motion/#usage> (дата обращения: 10.05.2019).
9. Википедия [Электронный ресурс]: Облачные базы данных — материал из Википедии — свободной энциклопедии: Версия 68907279/ Авторы Википедии // Википедия, свободная энциклопедия. — Электрон. дан. — Сан-Франциско: Фонд Викимедиа, 2016. — Дата обращения: 29.12.2016 /Режим доступа: Интерфейс пользователя
10. Википедия [Электронный ресурс]: Облачные базы данных — материал из Википедии — свободной энциклопедии: Версия 68907279 / Авторы Википедии // Википедия, свободная энциклопедия. — Электрон. дан. — Сан-Франциско: Фонд Викимедиа, 2016. — Дата обращения: 29.12.2016 /Режим доступа: User interface

11. Гаврилова Т А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. - СПб.: Питер, 2001.-384 с.
- 12.ГОСТ 19.502-78. Межгосударственный стандарт «Единая Система Программной Документации. Описание применения. Требования к содержанию и оформлению» от 01.01.1980 № 3350 // М.: Стандартинформ, 2010. с изм. и допол. в ред. от январь 2010
13. ГОСТ 19.503-79 Межгосударственный стандарт «Единая Система Программной Документации. Руководство системного программиста. Требования к содержанию и оформлению.» от 01.01.1980 № 74 // М.: Стандартинформ, 2010 с изм. и допол. в ред. от январь 2010
- 14.ГОСТ 19.504-79 Межгосударственный стандарт. «Единая система программной документации. Руководство программиста. Требования к содержанию и оформлению» от 01.01.1980 № 74 // М.: Стандартинформ, 2010 с изм. и допол. в ред. от январь 2010
- 15.ГОСТ Р 34.602-89 «Информационная технология. Техническое задание на создание автоматизированной системы» от 01.01.1990 // Стандартинформ. 2014 г. с изм. и допол. в ред. от 12.09.2018.
- 16.ГОСТ Р 19.201-78 «Единая система программной документации. Техническое задание. Требования к содержания и оформлению» от 01.01.1980 № 3351 // Стандартинформ. 2014 г. с изм. и допол. в ред. от 12.09.2018.
- 17.ГОСТ 34.602-89 Межгосударственный стандарт. «Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы» от июнь 2009 № 661 // М.: Стандартинформ, 2009 с изм. и допол. в ред. от 27.12.2006 г.
- 18.ГОСТ Р 52653-2006 Национальный стандарт Российской Федерации. «Информационно-коммуникационные технологии в образовании. Термины и определения.» от 01.07.2008 № 419-ст // М.: Стандартинформ, 2018 с изм. и допол. в ред. от 27.12.2006 г.

- 19.ГОСТ Р 52657-2006 Национальный стандарт Российской Федерации. «Информационно-коммуникационные технологии в образовании. Образовательные интернет-порталы федерального уровня. Рубрикация информационных ресурсов.» от 01.07.2008 № 423-ст // М.: Стандартинформ, 2007 с изм. и допол. в ред. от 27.12.2006 г.
- 20.ГОСТ Р 53620-2009 Национальный стандарт Российской Федерации. «Информационно-коммуникационные технологии в образовании. Электронные образовательные ресурсы. Общие положения" от 01.01.2011 № 956-ст // М.: Стандартинформ, 2018. с изм. и допол. в ред. от 15.12.2009 г
- 21.ГОСТ Р 56645.5-2015 Национальный стандарт Российской Федерации. «Системы дизайн-менеджмента. Термины и определения» от 01.06.2016 № 1577-ст // М.: Стандартинформ, 2016 с изм. и допол. в ред. от 19.10.2015 г.
- 22.ГОСТ Р 7.0.83-2013 «Система стандартов по информации, библиотечному и издательскому делу. Электронные издания. Основные виды и выходные сведения» от 15.10.2013 № 1163-ст // Стандартинформ. 2014 г. с изм. и допол. в ред. от 12.09.2018.
23. Интерфейс пользователя // Национальная библиотека им. Н. Э. Баумана Bauman National Library URL: [https://ru.bmstu.wiki/Интерфейс\\_пользователя](https://ru.bmstu.wiki/Интерфейс_пользователя) (дата обращения: 10.04.2019).
24. Информационно-управляющие человеко-машинные системы: Исследование, проектирование, испытания: Справочник/ А.Н. Адаменко. А.Т. Ашерев. И.Л. Бердников и др.; Под общ. ред. А.И. Тувинского и В.Г. Евграфова. - М.: Машиностроение. 1993. - 528 с.
25. Как мы держим наши гаджеты | Юзабилити // internet-tecnologies.ru URL: <https://www.internet-technologies.ru/articles/kak-my-derzhim-nashi-gadzhety.html> (дата обращения: 20.05.2019).
26. Мандел Т. Разработка пользовательского интерфейса / Пер.с англ. - М.: ДМК Пресс. 2008. - 416 с.

27. Мобильный UX дизайн: Основные принципы // UXgu URL: <http://uxgu.ru/mobile-ux/> (дата обращения: 20.05.2019).
28. Николаев А.Б. Интеллектуальные системы: учебное пособие / А.Б. Николаев, А.В. Остроух. - М.: МАДИ. 2012. - 271 с.
29. Оформитель библиографических ссылок // SNOSKA.INFO URL: <http://snoskainfo.ru/> (дата обращения: 14.02.2019).
30. Рыбина Г.В. Основы построения интеллектуальных систем. Учебное пособие. - М.: Финансы и статистика. 2010. - 432 с.
31. Сергеев С.Ф., Падерно И.И., Назаренко Н.А. Введение в проектирование интеллектуальных интерфейсов: Учебное пособие. - СПб: СПбГУ ИТМО. 2011. - 108 с.
32. Тема 13. Классификация пользовательских интерфейсов. Особенности не-графических пользовательских интерфейсов // studopedia.org URL: <https://studopedia.org/7-10486.html> (дата обращения: 11.04.2019).